University of Alberta Library
0 1620 2533828 4

For Reference

NOT TO BE TAKEN FROM THIS ROOM

Ex libris
universitatis
albertaensis

Digitized by the Internet Archive
in 2026 with funding from
University of Alberta Library

https://archive.org/details/0162025338284

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR David Arnold D'Amico

TITLE OF THESIS Contour Detection in Computed Tomography Images of Bone

DEGREE FOR WHICH THESIS WAS PRESENTED Master of Science

YEAR THIS DEGREE GRANTED 1986

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

THE UNIVERSITY OF ALBERTA

Contour Detection in Computed Tomography Images of Bone

by

David Arnold D'Amico

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF Master of Science

Department of Physics

EDMONTON, ALBERTA

1986

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled Contour Detection in Computed Tomography Images of Bone submitted by David Arnold D'Amico in partial fulfilment of the requirements for the degree of Master of Science.

Abstract

Bone disorders such as osteoporosis can be studied by quantitative analysis of computed tomography (CT) images. This involves precisely measuring the average volumetric density of trabecular bone which is non-homogeneously distributed throughout specific skeletal sites such as the wrist. The average trabecular bone mass is measured in regions of interest defined with reference to the bone contour. Thus, for precise evaluation of changes in bone mass as a function of time, highly reproducible detection of the bone contour is required to ensure that the same region of bone is analyzed each time.

Many techniques of contour detection for CT images suffer from sensitivity to artifacts and noise, and often they require user intervention or correction. Two new methods for automatic contour detection of bone in CT images of the forearm have been developed to reduce these problems: (1) 'cross-correlation of radial profiles', and (2) 'lower clipped skeletonization' (an operatation of Mathematical Morphology).

To evaluate the performance of these contour detection algorithms, they were applied to several test images, and the results were compared to other contour detection techniques. The test images included computer simulated objects with added blurring or noise, real CT images of the wrist which were artificially degraded to various degrees, and a large sampling of clinically acquired CT scans of the

forearm.

Compared with the Laplacian-filtering technique currently used in our Bone Research Unit, both new contour detection methods are more reliable and precise. In addition, 'edge detection by cross-correlation of radial profiles' is less sensitive to low or moderate levels of image noise than some of the best common contour detection algorithms. In a clinical application, 94% of the bone contours automatically determined by this method were judged to be acceptable.

Acknowledgements

Funding for this research was provided by the Natural Sciences and Engineering Research Council (NSERC) through a Postgraduate Scholarship and by the Alberta Heritage Foundation for Medical Research (AHFMR) through a Research Allowance. Their support is gratefully acknowledged.

I wish to express my sincere appreciation to my supervisors Dr. TN Hangartner and Dr. TR Overton for sharing their expertise, advice, and support throughout the project.

Special thanks is also extended to both Bob Morse and Bob Heath whose time and assistance prevented a lot of possible frustration. We enjoyed many interesting conversations which led to more intense investigations of and better solutions to several problems.

I would also like to thank all my other friends in the Department of Applied Sciences in Medicine and the Department of Physics who have made this experience most enjoyable and always remembered, notably Raman Paranjape and Derek Wells with whom much was discussed and shared, and Miles Ertman who also assisted with the photography.

Finally, I thank my parents and my wife Bonnie for their never-ending support away from the lab.

Table of Contents

Chapter Page

1. Introduction 1

1.1 Quantitative Measurement of Bone Density Using CT 4

1.2 Contour Detection 9

1.2.1 Pixel Classification 10

1.2.2 Edge Detection 11

1.2.3 Region Growing and Partitioning 13

1.2.4 Border Following 14

1.3 Two Newly Developed Compound Methods for Contour Detection of Bone in CT Images 15

2. The System of Bone Contour Detection Currently Used in our Bone Research Unit 18

2.1 The Pre-BOFI Stage of the BOFI System 18

2.2 The Auto-BOFI Stage 19

2.3 The Post-BOFI Stage of the BOFI System 20

3. Development of the New Contour Detection Algorithms 22

3.1 Image Analysis by Mathematical Morphology 22

3.1.1 Structuring Elements 23

3.1.2 Dilation and Erosion 26

3.1.3 Opening and Closing 31

3.1.4 The Hit or Miss Transformation 34

3.1.5 Sequential Thinning 36

3.1.6 Extension from Sets to Gray-Tone Images ...40

3.1.7 Contour Detection by the Algorithm of Lower Clipped Skeletonization (LCSKEL) 51

3.2 Cross-Correlation 59

3.2.1 Profiles for Cross-Correlation 60

3.2.2 Cross-Correlation and the Correlation Function 62

3.2.3 The Correlation Kernal 65

3.2.4 The Algorithm of Edge Detection by Cross-Correlation of Radial Profiles 70

4. Results of Comparing LCSKEL, EDCORP, and Other Contour Detection Algorithms 79

4.1 Image Set Based on an Ideal Square 80

4.2 Image Sets Based on Degraded CT Images of the Wrist 89

4.3 Image Set Representing a Clinical Study 99

5. Discussion 106

6. Conclusions 118

7. Future Developments 121

7.1 Areas of Refinement for EDCORP 121

7.1.1 Improving Profile Construction 121

7.1.2 Investigating Kernal Selection 123

7.1.3 Refinements to Contour Tracking 124

7.2 Investigations for LCSKEL 126

7.2.1 Threshold Selection 126

7.2.2 Combining LCSKEL With Other Methods 127

References 128

Appendix I 133

Appendix II 148

List of Tables

Table		Page
1.	Examples of structuring elements	25
2.	Values of R1 and R2 for the various algorithms applied to the images in the square image set	88
3.	Comparison of artificial and actual noise in the soft-tissue regions of distal and proximal CT images.	95
4.	Mean and standard deviation for bone overlay areas from the distal and proximal sets of images	99
5.	Comparison of the performance of EDCORP, LCSKEL, and BOFI system applied to a clinical image set	104

List of Figures

Figure Page

1. A typical set of CT images from a patient scanning session 7

2. Different types of edges 12

3. Dilation of a simple 1-D set X by structuring element B 28

4. Erosion of the 1-D set X in Figure 3 by the same structuring element B 30

5. The two steps of erosion and dilation leading to an opening (or closing) 33

6. The Hit or Miss Transformation of X with respect to B 35

7. Equivalent methods for computing the Hit or Miss Transformations in Figure 6 37

8. Two structuring element sequences for sequential thinning 39

9. The skeleton and the skiz resulting from sequential thinnings 41

10. The dilation and erosion of levels of a gray-tone image 43

11. The thinning of the gray-tone image X by the structuring element L 45

12. A 1-D example of the transformations incurred during lower skeletonization 48

13. A simple example of lower clipped skeletonization 50

14. Algorithm flowchart for contour detection by lower clipped skeletonization 52

15. Lower clipped skeletonization by LCSKEL 55

16. The lower clipped skeleton from Figure 15(c) after thresholding and then filling 58

17. Example of the profiles through a model radius 61

18. Example of a cross-correlation function 64

Figure Page

19. Emperical construction of the cross-correlation kernel 67

20. Algorithm flowchart for edge detection by cross-correlation of radial profiles 71

21. The edge detected by EDCORP 73

22. The definition of the edge point for ideal and ramp edges 83

23. Distal and proximal CT images of a wrist and determined radius overlays 94

24. Positive and negative pixel differences for the radius overlays of the 10 degraded images in the distal image set 96

25. Positive and negative pixel differences for the radius overlays of the 10 degraded images in the proximal image set 97

26. Plots of radius area versus axial location calculated from contours determined by BOFI and EDCORP 105

27. Different contour tracking directions and starting locations 109

28. A comparison of Rosenfeld's thinning and sequential thinning by Li for an image set with small holes 114

29. Possible one- and two-dimensional neighborhoods used in calculating the gradient 137

30. The eight compass directions for the Sobel compass gradient masks 139

1. Introduction

In many fields, image (or picture) processing is becoming as important as the development of equipment to obtain the pictures. In medical imaging, for example, image processing techniques aid in precise quantitative measurement of anatomical volumes and densities from images produced primarily for qualitative display. Generally, image processing is the manipulation and analysis of pictures by computer and includes the following areas [1]:

1. digitization and compression - sampling pictures to convert to discrete form; encoding or approximating images to save on computer system storage and communication.
2. reconstruction and restoration - reproducing images from sets of projections or encodings.
3. enhancement - improving degraded (blurred, noisy, low contrast) images.
4. recognition and description - segmenting images into parts which represent objects of interest within the picture; measuring features and properties (size, orientation, illumination, overlap) of these parts.

This last major branch of picture processing is referred to as image (or scene) analysis; its aim is to describe and quantify various aspects of objects within the image.

Before a picture can be analyzed it must be quantized or digitized. Usually, the (black and white) analogue picture, defined by an intensity or gray-level function

F(x,y) of two spatial coordinate variables in the image plane, is sampled and represented as a digital image X(i,j). The sample values are usually the average gray-level over small neighborhoods centered at rectangular grid points (x=i·d, y=j·d) where d is the sampling distance and i,j are integers. These sampled points or picture elements (pixels) are further quantized in terms of brightness using integral gray-level values. Typical imaging systems display digital images of size 256x256 pixels with 8 bit (256) gray-levels (0=black to 255=white). Many other sizes (64x64, 128x128, 512x512), gray-level scales (4, 16 bit), and sampling schemes (hexagonal, circular) for digital images exist, but these are less common. Once the data representing a digitized image is properly stored, it can easily be further manipulated and quantitatively analyzed via computer.

Image analysis, because of its quantitative and descriptive nature, is a very important activity in several fields. The following are a few examples of problems which require image analysis:

1. Mapping of specific types of terrain features (water, roads, forests, crops, urban areas) from digitized aerial photographs or *LANDSAT* images.
2. Computer recognition of hand writing or fingerprints.
3. Automatic detection and labelling of 'events' in digitized nuclear bubble chamber pictures.
4. Recognition of objects and formulation of plans of action from images 'seen' in robotics.

5. Evaluation of the mineral content in geological thin sections based on features (illumination, size distribution, texture) of regions within the digitized photograph, micrograph, or computed tomography image.
6. Precise quantification (determination, density, volume) of specific body structures (ventricles, vertebrae, tumors) from medical images (radioisotope images, digitized X-ray photographs, computed tomography images, nuclear magnetic resonance images).

In each of the above examples, the description or quantification of specific regions or objects within the image requires the isolation of these parts from the remainder of the picture or scene. An object (represented by regions of pixels of similar value or character) may be effectively isolated by marking its corresponding pixels in an overlay or mask image aligned with the actual picture. To this end, it is often necessary to obtain a precise and accurate contour separating the object from its surroundings. This is the goal of a major area in image analysis called image segmentation or contour detection, often loosely referred to as 'edge detection'.

Sometimes it is the contour itself which is important, as in radiation therapy treatment planning where the contours define the thickness of a region transversed by a radiation beam [13]. For other quantitative applications, the outline serves only to define a group of image pixels for further analysis, as in bone densitometry using computed

tomography (CT) images where the average bone mass interior to the bone contour is computed [14,15].

1.1 Quantitative Measurement of Bone Density Using CT

A significant application of contour detection has arisen in the field of medical imaging where it has become desirable to measure volumes, composition, and densities of anatomical structures and tissues *in vivo*. More specifically, to facilitate the diagnosis, study, and treatment of bone disorders such as osteoporosis, precise monitoring of bone mass from CT images requires a consistent method for detecting the contour separating the bone from the surrounding soft-tissue.

Osteoporosis is the most common of all bone dysfunctions in adults, and particularly in post-menopausal women, affecting about one in every four [16]. Osteoporosis is characterized by a decrease in bone mass (increased porosity) due to changes in the bone remodeling process (for example, the absorption rate of bone tissue exceeding the rate of bone formation to the point where spontaneous fractures can occur) [17]. The loss of bone mass is most apparent in the trabecular (spongy) bone, which is non-homogeneously distributed predominately in the vertebral column and the ends of long bones such as the tibia and the radius. The outer cortical (compact) bone shell, and especially its exterior (periosteal) surface, remains relatively unaffected. This is due to the much greater

surface contact that trabecular bone has with the metabolically important body fluids.

The normal rate of decrease of trabecular bone mass (TBM) in adults over the age of 35 years is 1 to 2% per year. Higher rates of bone loss (2 to 3% per year) can lead to osteoporosis. Consequently, a sensitivity of at least 1% is required for a procedure to monitor the progression or treatment of this disease [18]. Since the mass of bone varies at such a slow rate, even higher sensitivity is needed if long term studies are to be avoided. CT is the preferred method for studying changes in TBM since it is convenient, non-traumatic, provides true three-dimensional information on density distribution, and can be precise and accurate [15].

CT images are reconstructions of multiple X-ray attenuation measurements (profiles) made at various angles through the object [19]. The CT image is a digital, two-dimensional transaxial or cross-sectional view of a thin (1-10 mm) slice of the torso or appendicular site. Each pixel's CT number or gray-level value, which is proportional to the linear attenuation coefficient at that location within the object, can be related to local physical density if the material components of the object within the image are known.

The average density of the trabecular bone volume can be monitored by analyzing images produced by commercial whole body CT scanners, which generally have been designed

for qualitative display of large body sections. However, special-purpose CT scanners (usually with higher geometrical resolution due to a smaller field of view) have been developed to obtain images of peripheral sites such as the distal end of the forearm or the proximal tibia; analysis of these high resolution CT images leads to more precise evaluations of TBM [15,20].

Figure 1 shows a typical set of CT images of the distal forearm of a female subject. These images were produced by a specialized peripheral gamma-ray CT scanner [15]. The CT images of the forearm show the ulna (smaller cross-sectional area) and the radius, both surrounded by soft-tissue. Each bone consists of a shell of dense cortical bone filled non-homogeneously with porous trabecular bone. In adults, the space within the trabecular bone contains mainly fatty marrow. Because of its larger volume of trabecular bone, only the radius is considered for bone density measurements in our studies.

It has been found, for the measurement site at the distal end of the radius represented by the sequence of images in Figure 1, that the bone area changes by about 200 pixels (6%) per mm change in axial position due to the conical shape of the radius [15]. Thus, the cross-sectional area of the radius is taken as a sensitive indicator of axial position. The average trabecular bone density within a slice of the radius changes by up to 5% per mm variation in axial position [15]. Therefore, precise serial measurement

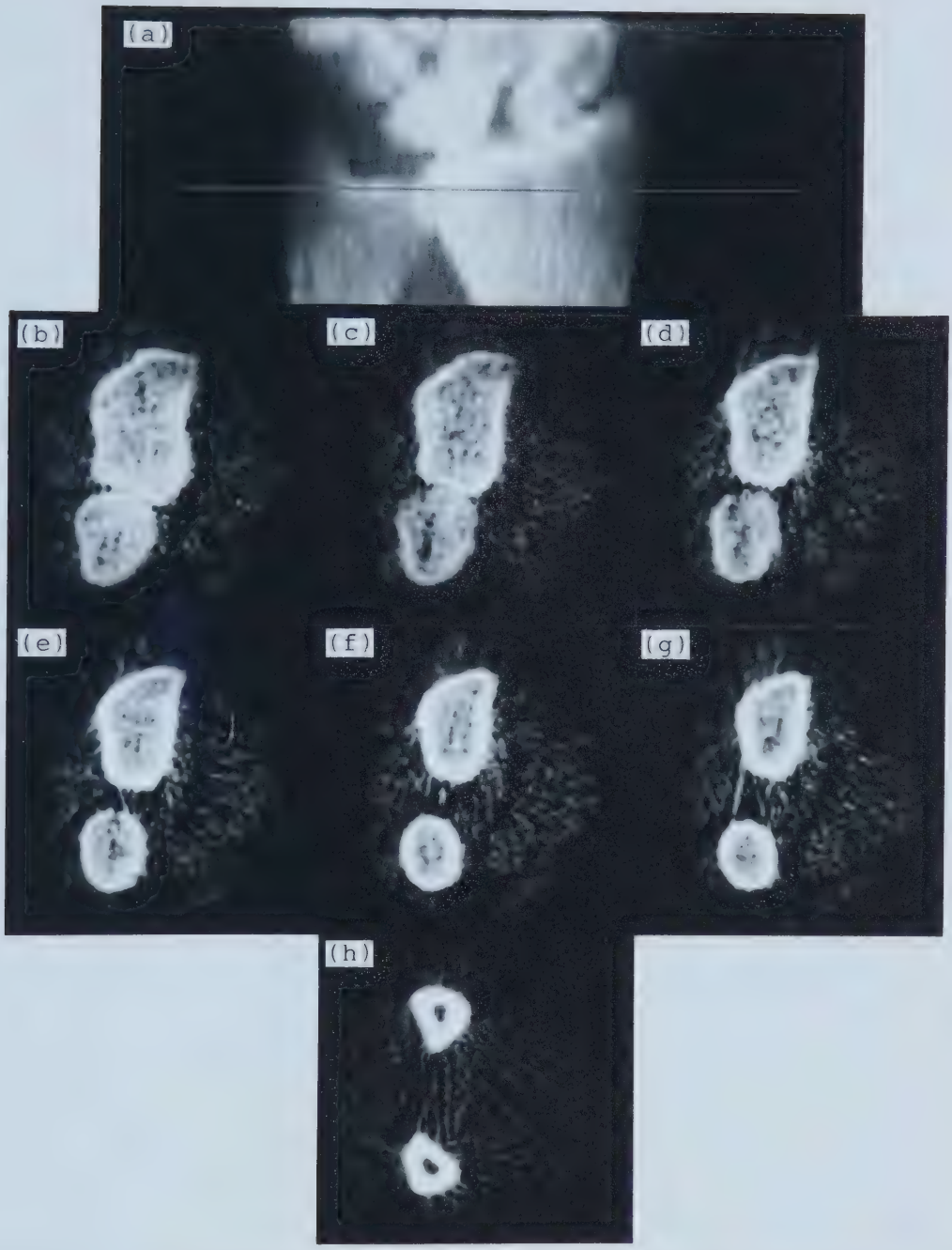


Figure 1. A typcial set of CT images from a patient scanning session. In each CT slice (b-h), the radius (larger) and ulna are visible. The slices (b) to (g) are at 2mm axial increments proximally from near the growth-plate of the radius indicated in the digital radiograph (scout scan) (a). Slice (h) is a one-third distal scan (measured from the wrist to the elbow).

of changes in bone mass over time requires highly reproducible detection of the contour of the radius to ensure that the identical volume of bone is evaluated each time. The outer edge of the cortex is normally chosen as a reference since it is not generally affected, over a period of a few years, by osteoporosis. In our studies, this bone contour is used to define a series of interior shells parallel to the bone contour over which the average bone density is computed. Usually, the interior 45% area arbitrarily defines the boundary between what is considered outer cortical and inner trabecular bone. This is a vague, low contrast boundary which precludes direct determination.

Note that a contour following the annular center or ridge of the cortical shell, would be everywhere roughly parallel to the outer edge of the radius. If the width of the cortical shell does not vary in time so that the central contour remains stationary, this contour instead could be used in serial studies to give a (slightly smaller) cross-sectional area suitable for indicating axial location and to define the trabecular regions of interest for TBM measurements.

For diagnostic studies where often a single measurement is used to classify and compare bone densities among groups of subjects, accuracy, as well as precision, is required. This is especially important if measurements of bone mass made at different laboratories are to be compared. However, repeated measurements to investigate changes in bone mass in

an individual subject are likely more useful due to the wide variation in normal bone mass among adults of even the same age group and sex. Therefore, precision in all steps, including contour detection, is more important than accuracy.

Since detection of the contour must be highly reproducible, operator intervention should be kept to a minimum. Automation also reduces the time required for data analysis since less time is needed to guide the algorithm or to correct errors in the detected contour. In addition, if the contour detection algorithm is sufficiently fast (under a minute), it can be used promptly after the first patient scan has been completed; the operator may then use the computed cross-sectional bone area as an aid in positioning the next measurement slices. This could reduce the number of scans and the total time required per patient scanning session, thereby reducing the demands made upon the patient for immobility during a measurement series.

1.2 Contour Detection

To obtain a precise and accurate contour separating a particular object from the rest of the image, contour detection may be achieved by different methods. These include pixel classification, edge detection (in the true sense), region growing or partitioning, border following, or combinations of these.

1.2.1 Pixel Classification

Pixel classification is an attempt to classify pixels on the basis of some measurable property. If the probability of classes is known along with the probability density of the values for each class, then each pixel can be classified based on criteria which minimize the expected classification error.

A very common property for pixel classification is gray-level value. By thresholding the gray-levels of an image, the pixels are being classified as lighter or darker. For example, a gray-level histogram of an image should display peaks corresponding to gray-level ranges occupied by various objects within the image. If these peaks do not overlap too greatly, then they can be separated by choosing thresholds at the gray-level values corresponding to the valleys between them. The pixels with gray-levels between these thresholds are identified and marked depending on their gray-level. The various objects can then be displayed approximately segmented by the chosen marking scheme.

Other local properties besides the gray-level value (such as pixel value difference, gradient magnitude, local averages) or 'typicality' values based on statistics of the entire image may be used in pixel classification. Scatter plots together with histograms can be used to visualize clusters of object pixels in multi-dimensional property spaces.

1.2.2 Edge Detection

In a gray-level image, an untextured edge is defined as the boundary between two regions of different average gray-levels [21]. There also exist textured edges which are boundaries between regions of different texture [22]. The edge beween two such regions may be very conspicuous even if their average gray-levels are identical. For example, one region may be homogeneously mid-level gray while the other region contains pixels half of which are black and half white, randomly interspersed. Usually, another image descriptor or local property other than gray-level is used to characterize such regions [2]. The value of this local property can be thought of as the 'gray-level' of the image in some texture space so that the above definition of an untextured edge can be applied to the transformed image. For example, the 'dottedness' of a neighborhood may be transferred to a gray-level by applying Laplacian like operators [2].

The most common type of (untextured) edge in a theoretical sense is the step edge shown in cross-section in Figure 2(a). Blurring or defocusing (finite photon beam width, reconstruction error) and noise (electronic, photon statistics) are factors which degrade these step edges so that they actally appear in images as noisy ramp edges as shown in Figure 2(c). The incline of the ramp defines the sharpness or extent of the edge. Edges, even within a single image, will generally vary in quality (freedom from noise)

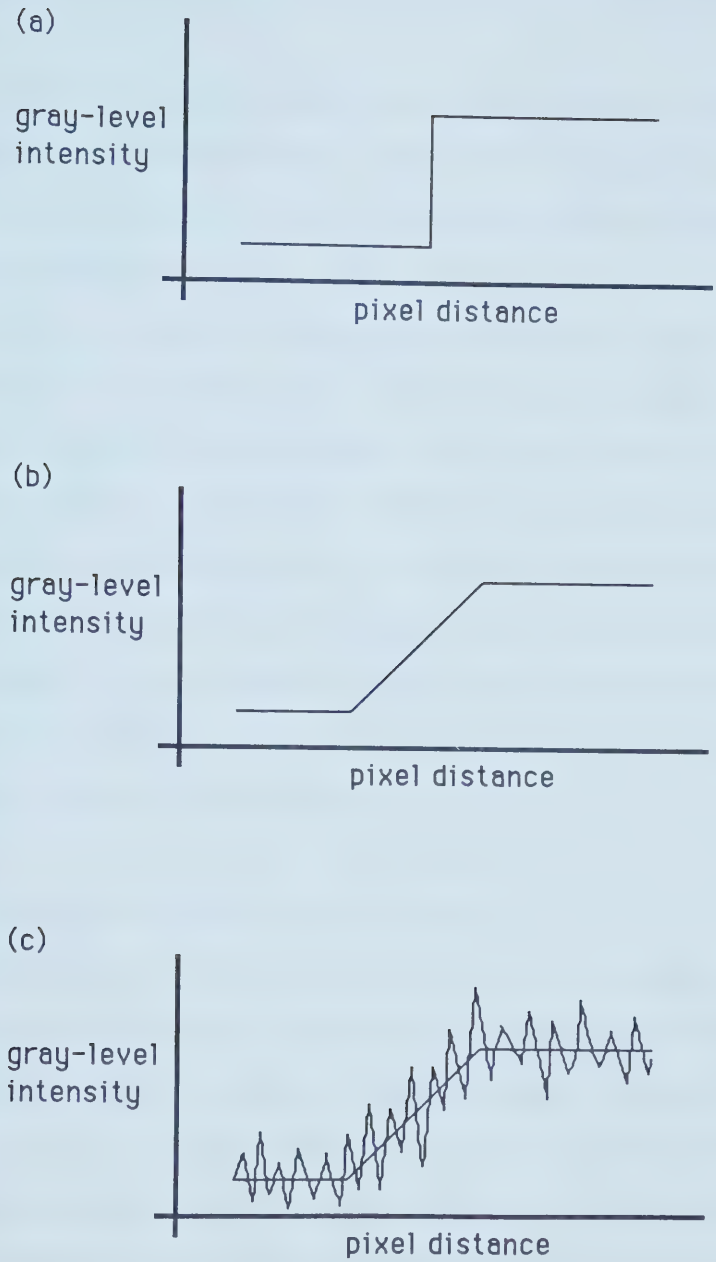


Figure 2. Different types of edges: (a) Ideal step edge (in cross-section orthogonal to the direction of the edge); (b) Ramp edge due to blurring of step edge; (c) Ramp edge with noise.

and sharpness; so detecting them becomes a difficult problem.

Most edge detection techniques are special cases of pixel classification methods. Many involve specific applications of digital spatial frequency filtering. The most common edge detection techniques are based on local differencing, estimating the gradient at each pixel by using approximations to directional derivatives. An edge point is often classified by thresholding the value produced by such local differencing operations. Other edge detection techniques use very different classification procedures. Some try to find the 'best fitting' edge function in regions of the image; some attempt to follow or track edges through the image. Details of various types of edge detection methods are given in Appendix I.

1.2.3 Region Growing and Partitioning

Region growing [3] can be similar to non-parallel edge tracking procedures (see Appendix I). Rather than searching for a path through the image based on edge strength and direction, small neighborhoods or cells of pixels are examined and added to the current region if they meet certain homogeneity criteria. These criteria are based on the nature of the region or the candidate cells. For example, acceptance of a cell may depend on the similarity of its average gray-level compared to the region's, its closeness to the region, and on the size and shape of the

current region.

Region partitioning [4] uses the same methods as region growing to allow both merging and/or splitting of many regions. Several cells or regions are considered, and adjacent pairs are merged (or split) if they are mutally compatible (non-compatible).

1.2.4 Border Following

Border following procedures [23,14] are very simple methods. They usually define a 'world model' to aid the method in determining specific regions of the image. Often, this world model is defined by gray-level thresholds; a pixel belongs to the region or object if its value is within the threshold window. Note that this is just a simple pixel classification scheme. The algorithm then attempts to follow the border of this region producing a (closed) contour. The process uses the 'last few' previously determined border pixels to define a border direction. A search is conducted in this general direction for the next border pixel belonging to the object. The 'last few' previous border pixels are updated, and the recursive search for new border pixels is repeated, usually until the contour connects with itself. Local filtering operations may be incorporated into the border following procedure to reduce the effects of noise [14].

1.3 Two Newly Developed Compound Methods for Contour Detection of Bone in CT Images

Many contour detection methods are very general in design, though some can be tuned to particular applications or classes of images. If a contour detection method is designed for general application on many different kinds of images, then it will usually enhance or determine all contours in the image. Its generality for all image classes implies its lack of selectiveness for subsets of contours within an image. Thus, for specific contour detection in a single class of images, such as detection of the edge of the radius in CT images of the wrist, a contour detection algorithm should be tuned to perform optimally on contours with those particular characteristics.

With this consideration, a modular, fully automated, edge detection algorithm was developed for determining the outer edge contour of the radius in CT images of the forearm. 'Edge detection by cross-correlation of radial profiles' (EDCORP) combines several aspects of image analysis including pixel classification using thresholding, edge matching or filtering, edge tracking (with backtracking), and creating an overlay with smoothing applied. With small changes, however, the general formulation of EDCORP is applicable to many other classes of images. In fact, a similar algorithm was developed independently by Pope *et al.* for detection of the left ventricular boundary in cine-angiocardiography images [24].

Using methods very different from those of EDCORP, a unique, semi-automated algorithm for isolating the radius in CT images was developed based on a 'ridge' enhancing technique in combination with pixel classification by thresholding. Using certain operations from Mathematical Morphology, a theory of image analysis developed by G. Matheron and J. Serra [25], the central ridge of the cortical shell of the radius is determined by an algorithm called 'lower clipped skeletonization' (LCSKEL). In LCSKEL, the operation of 'skeletonization' reduces all ridges in an image to single-pixel widths, while at the same time suppressing the background. The Mathematical Morphology operation of 'end-point clipping' retains only closed ridge contours. Thresholding then attempts to maintain only the main cortical ridge. The operations of 'opening' and 'closing' are used to smooth the radius overlay once it has been created by filling the determined ridge contour. With modifications, the methods of LCSKEL could be applied to many other contour extraction problems.

An important but difficult job in image analysis is to evaluate the performance of contour detection techniques. The results from EDCORP and LCSKEL were compared with each other and with other contour detection algorithms when applied to various test images. The other contour detection algorithms included some of the best local edge detection methods in the literature as well as a semi-automated contour detection algorithm previously designed for

outlining the radius in CT images of the forearm. The test images included special model images with added blurring or random noise and real CT images of the wrist that were artificially degraded to various degrees. Of course, the best test of our contour detection methods is in the clinical application for which they were designed. Therefore, a few hundred patient CT scans were used to compare EDCORP and LCSKEL with the method of contour detection currently used in our Bone Research Unit.

2. The System of Bone Contour Detection Currently Used in our Bone Research Unit

The BOFI system (BOne FInd) is currently used to segment the radius from the CT image for further analysis. This system is composed of a Pre-BOFI image preparation stage, the actual contour detection stage of Auto-BOFI, and a Post-BOFI viewing and correction stage which also submits the final segmented image for bone area and density measurements.

2.1 The Pre-BOFI Stage of the BOFI System

In Pre-BOFI, the 256x256 CT image of the forearm is displayed on a Grinnell imaging system. A trackball and cursor system is used by the operator to frame a 128x128 pixel subimage roughly centered on the radius bone. In framing the radius, the operator tries to exclude the ulna; if this is not possible, the ulna must be 'erased' by drawing a line separating the ulna from the radius using the trackball and cursor. All pixels along each column below this separation line are replaced with a single value equal to that of the pixel at the intersection of this column and the line. Of course, this assumes the separation line is drawn through soft-tissue and that the orientation of the image is such that the ulna is roughly below the radius as in Figure 1. This erasing process effectively replaces the ulna with soft-tissue and prevents the ulna from interfering with the detection of the radius contour.

2.2 The Auto-BOFI Stage

Pre-BOFI prepares the CT image for the actual contour detection by Auto-BOFI (or just BOFI). Auto-BOFI uses Laplacian filtering in conjunction with median filtering to enhance the bone edge.

The Pre-BOFI subimage containing only the radius is first 9x9 median filtered to reduce the effects of noise while preserving the sharpness of the edge [27]. Each pixel in the image is replaced by the median of the 81 pixels in the 9x9 square centered at that point. Such filtering is required since the Laplacian is extremely sensitive to noise [5].

The Laplacian operator (see Appendix I) is then convolved with this median filtered subimage (a 9x9 pixel Laplacian was found to work best with 9x9 median filtering). The resulting Laplacian image is an approximation to the two-dimensional second derivative at each point. Where the image is of constant value (no edges), the Laplacian is zero. In the proximity of an edge, the Laplacian value is positive for points just before or below the edge and negative for points just after or above. At the edge point, the Laplacian value crosses through zero, and these are the points BOFI seeks.

Each row of the Laplacian image is scanned point by point until a negative threshold is exceeded negatively. The algorithm then backs up a few pixels from this point, into the pixels just before the edge with positive Laplacian

value. Forward scanning resumes but this time a zero threshold criteria is invoked. When this zero threshold is then negatively exceeded, Auto-BOFI has found the zero-crossing of the Laplacian signifying a detected edge point. The negative threshold is chosen low enough to minimize the chance of selecting false edges due to noise and high enough so that the true cortical edge is not by-passed in the scanning.

This scanning is repeated for each row, both from the left and the right, and then for each column, from the top and the bottom. If a point is incorrectly chosen as an edge point during a row scan, the column scan will likely correct the mistake. During scanning, all pixels of the original image which are outside (below) the edge point are negated; this marks the soft-tissue and creates, essentially, a radius overlay image. The resulting 'strip-negated' subimage, as it is called, effectively isolates the radius from the rest of the CT image.

2.3 The Post-BOFI Stage of the BOFI System

In Post-BOFI, the subimage is displayed with the bone contour superimposed to allow the operator to visually assess the contour. Because of the high sensitivity of the Laplacian operator to noise, the automatically determined contour from Auto-BOFI may miss portions of the visually apparent edge contour. Thus, a corrected contour must be 'drawn in' by the operator using the cursor and trackball

system. Because this visual correction is very subjective, reproducibility is greatly reduced. Besides not correctly detecting portions of the radius contour, small closed contours outside the radius are sometimes produced by Auto-BOFI. The operator must then 'select' the actual bone contour so that Post-BOFI can delete these other false objects. Manual contour correction and selection greatly increases the time and effort required by the operator to process the CT images.

The final strip-negated subimage is updated with the contour corrections and is then submitted for density analysis. For bone mass measurement, a series of shells parallel and interior to the radius contour are defined; the average volume density over each shell, together with the pixel area of that shell, are computed and stored in a data pool.

In later processing, the inner 45% of the total bone area and the average trabecular bone mass (TBM) over this region are computed from the shell data for each CT slice. For each scanning session, a plot of the TBM versus axial slice position is produced, and a smooth curve is fitted through these plotted points. This allows a volumetric trabecular region of interest for the wrist to be defined; the average TBM over this volume is calculated for each measurement date. Finally, from a series of such volumetric density measurements for varios dates, a plot of the patient's TBM versus time is produced.

3. Development of the New Contour Detection Algorithms

The two methods of contour detection which were developed for automatic isolation of the radius in CT images of the wrist are (1) 'lower clipped skeletonization' (LCSKEL) through image analysis by Mathematical Morphology; and (2) 'edge detection by cross-correlation of radial profiles' (EDCORP). EDCORP was developed mainly from concepts common to digital signal processing. Both algorithms, however, share the morphological operations of 'opening' and 'closing' which are based on 'erosions' and 'dilations' by structuring elements. LCSKEL, in addition, employs more involved but related morphological transformations. Thus a limited knowledge of the theory of Mathematical Morphology is required prior to discussion of either contour detection algorithm, and some concepts of cross-correlation are reviewed before discussing EDCORP.

3.1 Image Analysis by Mathematical Morphology

Mathematical Morphology is a set theory developed by G. Matheron and J. Serra in an attempt to provide a coherent framework for image processing. Morphology is the study of form or structure; the main objective of Mathematical Morphology is to reveal structures in images by performing sequences of neighborhood transformations on sets which model these images [25]. Most of these neighborhood transformations are based on combinations of erosions and dilations by various structuring elements together with the

operations of union, intersection, and set difference. By choosing the appropriate structuring elements and applying the combinations of operations in the proper sequence, a wide range of enhancements or quantifications of the image are possible [25].

Of the many transformations available in Mathematical Morphology, only opening, closing, lower thinning, and end-point clipping were used. Lower thinning and end-point clipping are based on the 'Hit or Miss transformation' which, like both opening and closing, is built up from erosions and dilations. Thus, all four transformations can be programmed as combinations of erosions and dilations; however, more computationally efficient algorithms were developed when required.

Initially, it is easier to understand these Mathematical Morphology transformations as applied to 'sets' of pixels or, equivalently, binary images where pixels are either 'on' or 'off'. The extension to operating on gray-tone images then easily follows.

3.1.1 Structuring Elements

The basic tool of Mathematical Morphology is the structuring element. The structuring element, together with a morphological operation, determines what type of quantification or structure in the image will be emphasized by the transformation. Even with a given operation, the different transformations resulting from employing different

structuring elements will allow for very different measurements or enhancements of the image. The structuring element is much like a kernel or filter in two-dimensional signal processing. Usually, the structuring element is translated through the image pixel by pixel in analogy with 2-D linear convolution of an image with a kernel. The actual mathematical operation of convolution (the summation of the products of the overlayed pixels in the image with the overlaying points in the kernel [28]) is replaced with the morphological operation being performed. At each pixel location, a result is written to the corresponding location in the output image. This resultant image may then be further transformed or quantified.

Table 1 shows several examples of structuring elements and some operations which employ them. For binary images, structuring elements test whether certain pixels in the image are 'on' (non-zero) or 'off' (zero). The test for an 'on' pixel is denoted by (•) while a test for an 'off' pixel is (o). If the structuring element will accept a match with either 'on' or 'off' pixels, this is denoted by (·). Useful quantification of the resultant images and/or the structures which are emphasized by the transformation are also listed in Table 1. For example, when the structuring element A is used by the Hit or Miss transformation applied to a binary image, a test is made at each point for an 'on' pixel, and a non-zero value is written to the corresponding pixel in the resultant image if the test is true. Note that the result is

Table 1. Examples of structuring elements, their associated morphological operations, and the effects of the resulting image transformations.

Symbol	Structuring Element	Morphological Operation	Quantification Parameter	Structural Effect
A	•	dilation	pixel area	identity transform
BC3	• · · · •	erosion	set covariance size=3	
BS	o •	Hit or Miss	number of end-points	location of end-points
BS3	o • • • o	Hit or Miss	linear size frequency l=3	chord l=3 location
S	• • • • • • • • •	opening closing		smoothing (filtering) (filling)
L	o o o · • · • • •	thinning		homotopic skeleton
E	o · o o • o o o o	thinning		end-point clipping

simply an identity transformation. The number of 'on' pixels in the resultant image is then a measure of the images's pixel area.

Further examples of structuring elements and how they are designed will be described as some of the operations and transformations of Mathematical Morphology are introduced.

3.1.2 Dilation and Erosion

The two basic morphological operations from which more complicated transformations can be built are dilation and erosion. This is in analogy with arithmetic operations of a computer, such as multiplication and division, being based on addition and subtraction.

Dilation of a set of pixels X by a structuring element B is denoted by:

$$Y = (X \oplus \check{B}).$$

X refers to the set of those binary pixels which are 'on' (•). $\check{B}$ is the transpose of B; that is, a reflection through the origin of the structuring element. This is similar to a transpose of the kernel in 2-D linear convolution [28]. An example of a structuring element B and its transpose $\check{B}$ are shown at the top and bottom, respectively, of Figure 3. The dilated set Y is defined as the locus (union) of origins (x,y) of all B_{xy} that hit (↑) or intersect ($\cap$) the set X:

$$Y = (X \oplus \check{B}) = \{(x,y): B_{xy} \uparrow X\}$$
$$= \{(x,y): (B_{xy} \cap X) \neq \emptyset\},$$

where (x,y) denotes a point or pixel in the frame or matrix of the image and B_{xy} denotes a translate of B such that B has its origin located at point (x,y). For B_{xy} to hit the set X, at least one 'on' pixel of B must overlap a ('on') pixel of the set X.

The above operational definition of the dilation is equivalent to the mathematical definition:

$$Y = (X \oplus \check{B}) = \bigcup_{mn \in \check{B}} X_{mn}$$

which states that the dilated set Y is the union (U) of the sets X translated by m,n where (m,n) is an element of the transpose of B ((mn) ϵ $\check{B}$). In Figure 3, a simple 1-D set is dilated by a structuring element B using both definitions. Notice that dilation, as the name implies, causes an expansion of the set X by B.

In most cases, the operational definition is preferable since it allows for easier visualization of the transformation. The operational definition also relates the dilation to linear convolution; in both cases, we translate the structuring element (kernel) through the input image and at each pixel assign a value to the corresponding pixel of the output image. In convolution, a transposition of the kernel is implied. In dilation, the notation shows a transposed structuring element, but the operational definition does not use the transposition. Note that the dilated set Y occupies the same 'region of support' [28] as would convolution of X by a kernel $\check{B}$ using ones and zeros in place of •'s and ·'s.

Unlike dilation, erosion has no analogous signal processing operation. Erosion of a set X by a structuring element B is defined as the locus of origins (x,y) of all B_{xy} that are completely included in the set X:

$$Y = (X \ominus \check{B}) = \{(x,y): B_{xy} \epsilon X\}.$$

This operational definition for the eroded set Y is supplemented by an equivalent mathematical definition:

$$Y = (X \ominus \check{B}) = \bigcap_{mn \epsilon \check{B}} X_{mn}$$

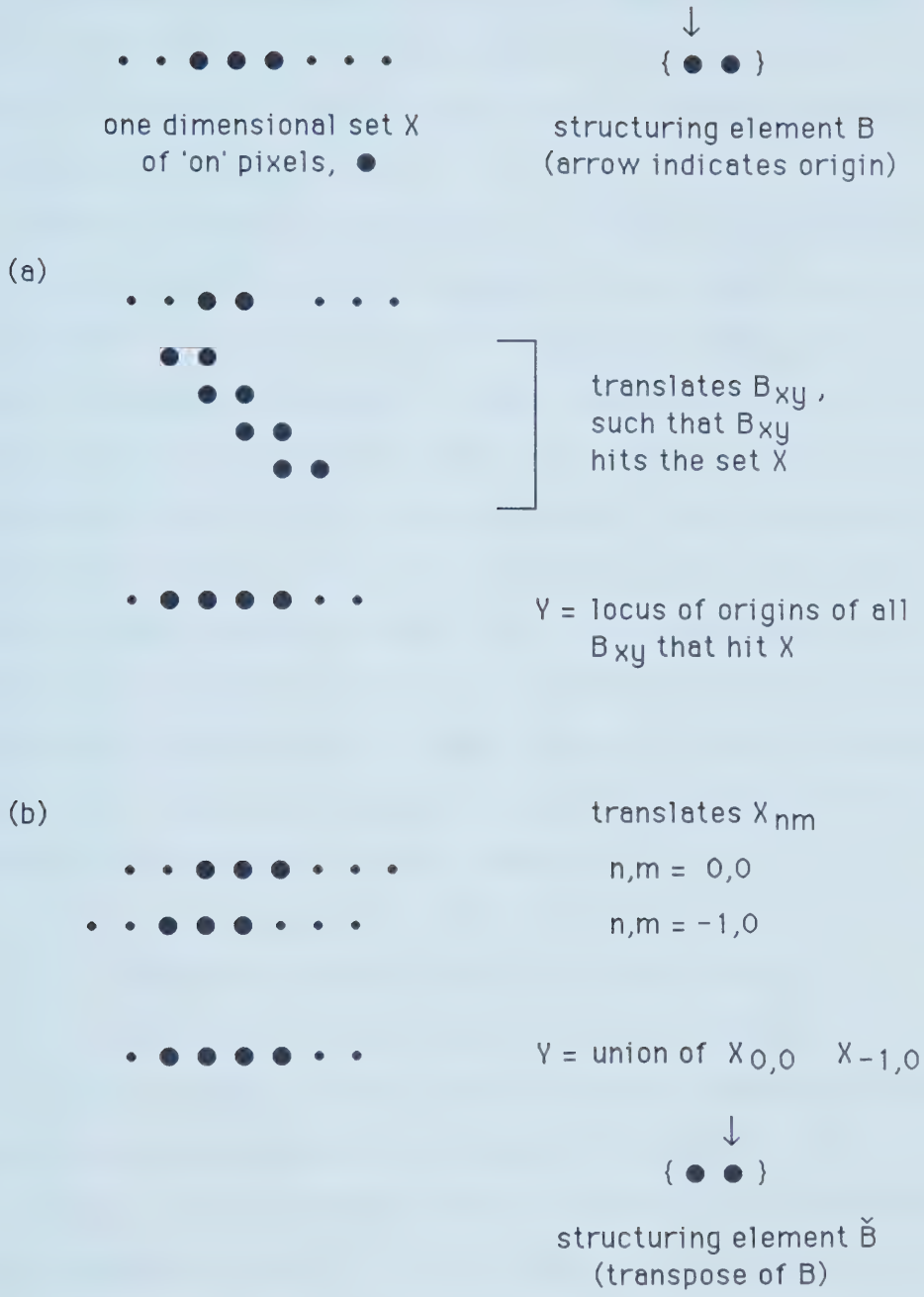


Figure 3. Dilation of a simple 1-D set X by a structuring element B: (a) Dilation using the operational definition; (b) Dilation using the mathematical definition.

where $\cap$ denotes intersection. Figure 4 shows the erosion of the set X by the structuring element B, both from the example in Figure 3. As the name implies, erosion causes a shrinking of the set X by B.

Several algebraic properties of erosion and dilation are given by Serra [25]. For example, from inspection it can be seen that dilation and erosion dual each other with respect to complementation. That is, complementing the erosion of X by B results in the same image as dilating the complemented set X^c by B, and conversely. (The complement of a set is the set of 'on' pixels resulting when all 'off' pixels in the image are turned 'on' and *vice versa*.) Also, the erosion or dilation of X by a large or complex B can be performed by decomposing B into a dilation (sum) of smaller, simpler B_i's and then eroding or dilating the intermediate erosions or dilations with the component B_i's:

$$\text{Suppose } B = (B_1 \oplus B_2) \text{ then}$$

$$(X \ominus \check{B}) = X \ominus (\check{B}_1 \oplus \check{B}_2) = (X \ominus \check{B}_1) \ominus \check{B}_2$$

$$(X \oplus \check{B}) = X \oplus (\check{B}_1 \oplus \check{B}_2) = (X \oplus \check{B}_1) \oplus \check{B}_2.$$

The complete proofs of this iterativity and the duality with respect to complementation are given in Appendix II.

Another property dealt with in Appendix II is that of local knowledge. Often, it is assumed that the set X is known only inside a frame or window Z. It seems reasonable that when the image Z is acted upon by a structuring element, in either erosion or dilation, the frame of the result is reduced in size (eroded) by that structuring

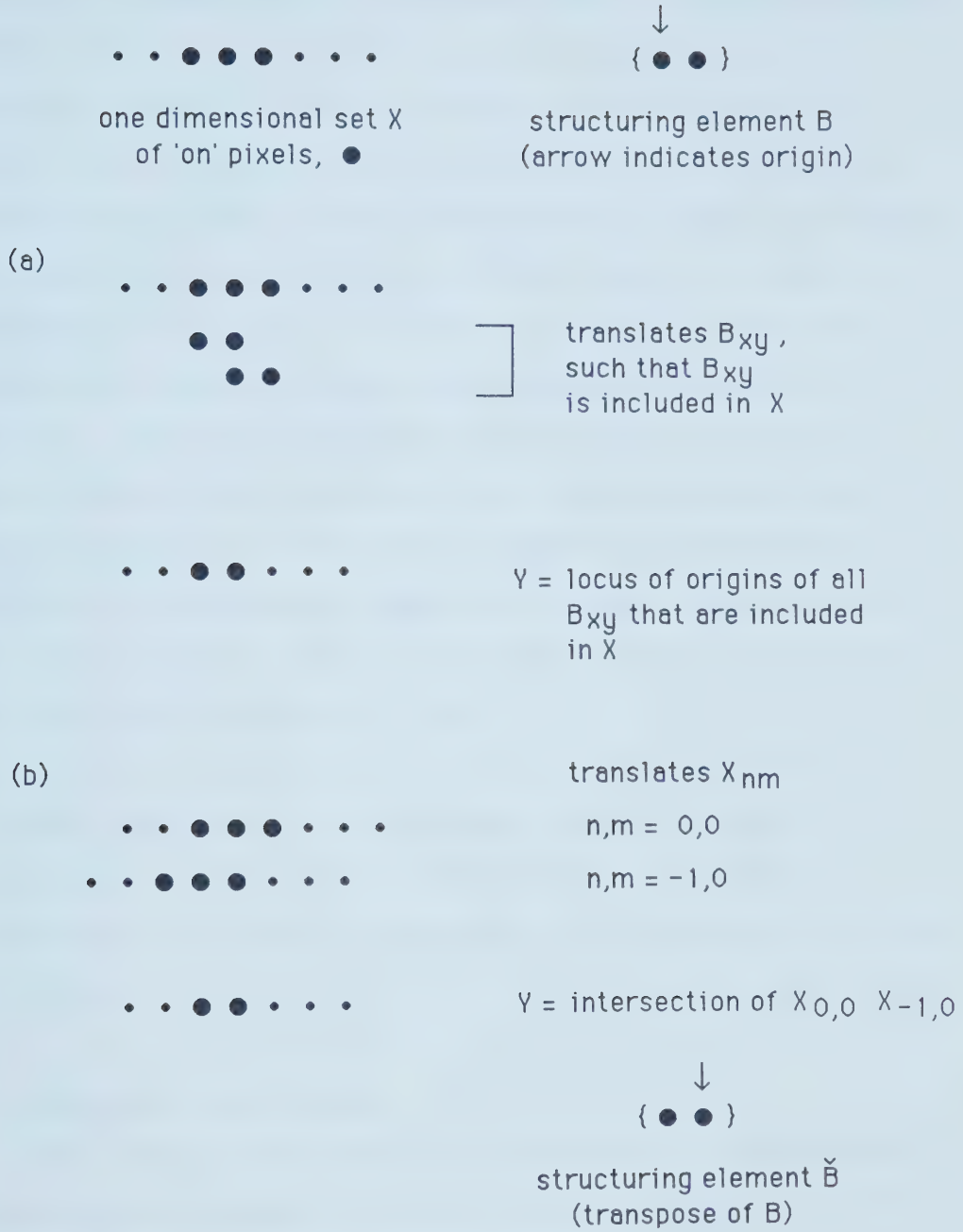


Figure 4. Erosion of the 1-D set X in Figure 3 by the same structuring element B: (a) Erosion using the operational definition; (b) Erosion using the mathematical definition.

element as exemplified in Figures 3 and 4. This is analogous to 2-D linear convolution where the border pixels of the image are commonly ignored (zeroed). If the convolution kernel was 5x5 pixels, the 256x256 result of filtering a 256x256 image would contain zero-definite pixels in a border two pixels wide on each side. Alternatively, it could be assumed that the set X is completely known and that the background outside of the frame extends (as zeros) to infinity. The frame would then be considered infinite in extent. This latter assumption will be maintained in all further examples of Mathematical Morphology; therefore, erosion of the infinite frame has no effect on the image. Only the required portion of an image will be illustrated in all additional examples.

With these properties in mind, combinations of erosions, dilations, intersections, unions, and set differences can be used to create other morphological transformations which are employed by our methods of contour detection.

3.1.3 Opening and Closing

The Mathematical Morphology operations of opening and closing were employed by both methods of contour detection to smooth the final radius bone overlays.

The opening of set X with respect to a structuring element B is denoted by:

$$X_B = (X \ominus \check{B}) \oplus B.$$

After eroding X by B, it is not generally possible to recover X by dilating the result with the same (but transposed) structuring element. Instead, the dilate reproduces only a simpler, less detailed subset of X. The opened set X_B is the domain swept out by all the translates of B which are completely included in the set X. Geometrically, this transformation smooths the contours of X, cutting narrow isthmuses and suppressing small islands and capes. An example of an opening is given in Figure 5. The two step process of erosion followed by dilation can be conceptualized as a single step process of setting to 'on' all pixels in each and every translate of B which is able to fit completely inside X. Notice how opening performs somewhat as low pass filtering, removing the noisy (small with respect to the structuring element) components of X.

The complement of opening is closing. The closing of set X with respect to B is the following set:

$$X^B = (X \oplus \check{B}) \ominus B.$$

Closing the background X^C of X is equivalent to complementing the opening of the set X, and conversely:

$$\begin{aligned}(X^C)^B &= (X^C \oplus \check{B}) \ominus B \\ &= (X \ominus \check{B})^C \ominus B \\ &= [(X \ominus \check{B}) \oplus B]^C \\ &= (X_B)^C.\end{aligned}$$

The closed set X^B is the set of pixels (x,y) such that all the translates of B which contain (x,y) also hit X. This transformation blocks up narrow channels, small lakes, and

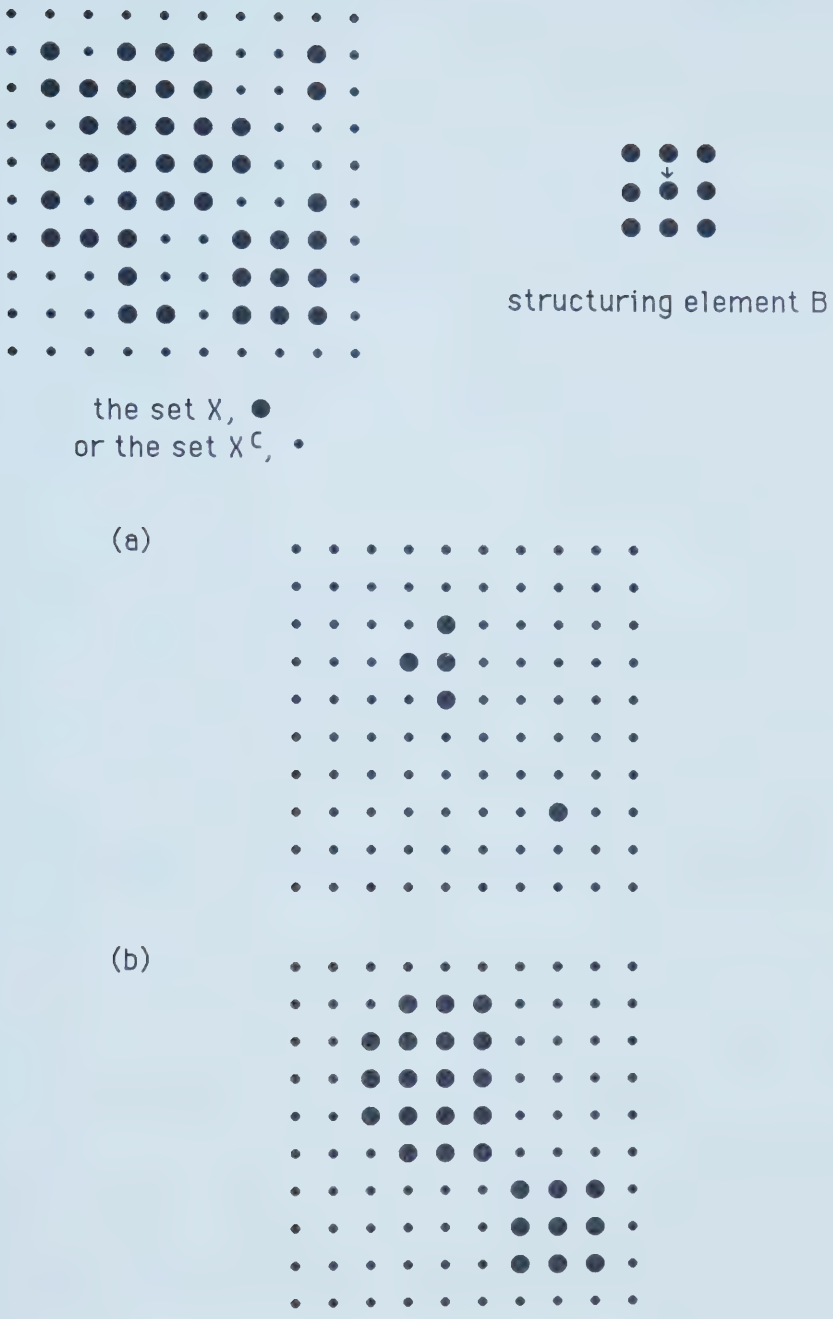


Figure 5. The two steps of erosion and dilation leading to an opening performed on the set X. Also, the two steps of dilation and erosion leading to a closing when performed on the background X^C of set X: (a) $(X \ominus \check{B})$ or $(X^C \oplus \check{B})$; (b) X_B or $(X^C)^B$.

long thin gulfs. Figure 5 also illustrates the closing of X^C by B where X^C is considered to be the set of 'off' (·) points.

While the opening X_B of X with respect to B is completely contained within X, the closing X^B completely contains X. The opening smooths X by filtering out small protrusions into which the structuring element cannot fit; the closing smooths X by filling small intrusions which B is able to bridge or span.

3.1.4 The Hit or Miss Transformation

The Hit or Miss transformation employs what can be considered a two-part structuring element B composed of B^1 (•) and B^2 (o). A point (x,y) belongs to the Hit or Miss transform $(X \circledast \check{B})$ of X, if and only if the translate B^1_{xy} is completely included in X and B^2_{xy} is completely included in the complement X^C:

$$(X \circledast \check{B}) = \{(x,y): B^1_{xy} \subset X;\ B^2_{xy} \subset X^C\}.$$

Note that if B^2 is empty so that B consists only of B^1, then the Hit or Miss transform reduces to an erosion.

An example of the Hit or Miss transform using the above definition is given in Figure 6. The Hit or Miss transform can instead be computed using combinations of erosions and dilations:

$$\begin{aligned}(X \circledast \check{B}) &= (X \ominus \check{B}^1) \mid (X \oplus \check{B}^2) \\ &= (X \ominus \check{B}^1) \cap (X^C \ominus \check{B}^2).\end{aligned}$$

The first equation says to erode X by B^1, dilate X with B^2,

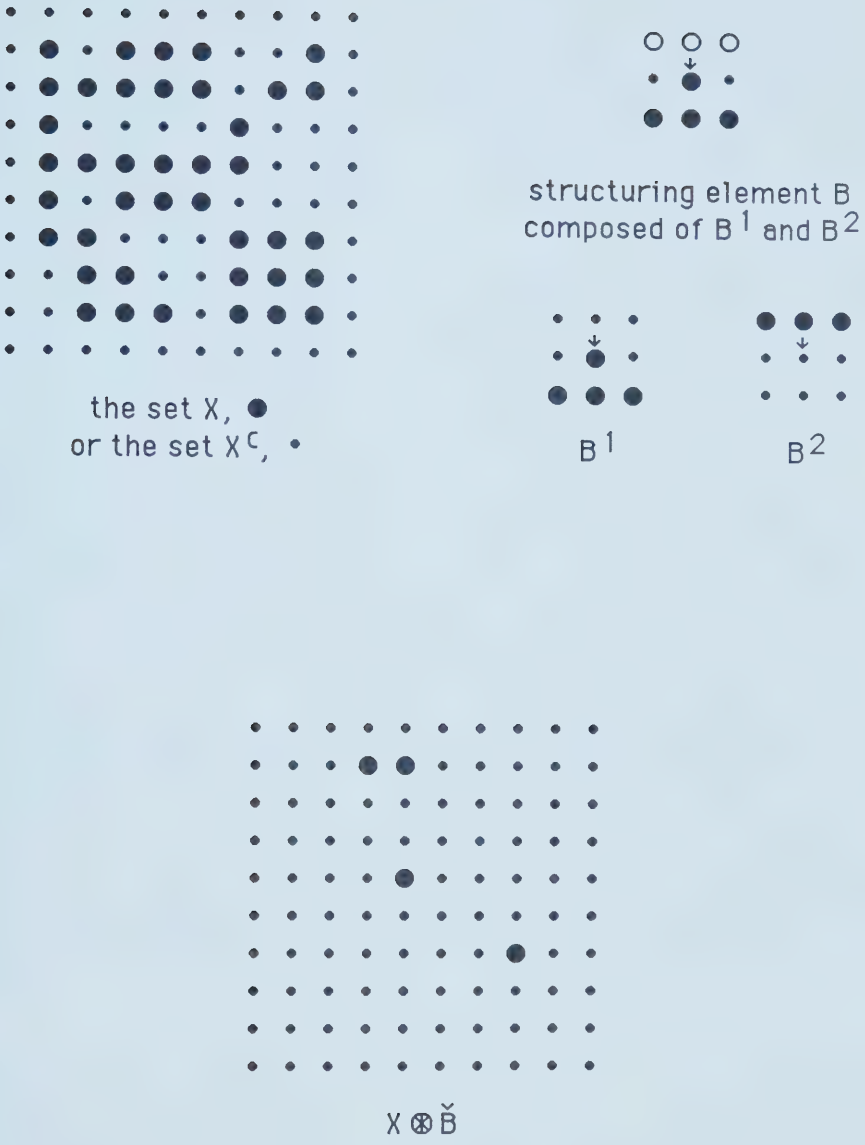


Figure 6. The Hit or Miss transformation of X with respect to B.

and then perform, the set difference of the eroded set less the dilated set. (A set difference Y|Z is the set of all 'on' pixels in Y less any corresponding 'on' pixels in Z; pixels in Z which are not in Y are ignored.) The use of this definition is illustrated in Figures 7(a) and (c) where X and B are the same as in Figure 6. The second equation is exemplified in Figures 7(b) and (c). Here, the intersection of the intermediate results of X eroded by B^1 and X^C eroded by B^2 gives the Hit or Miss transform of X. Note that the duality of erosion and dilation with respect to complementation can be employed if desired.

The Hit or Miss transform is an example of a more involved Mathematical Morphology operation which can be reduced to combinations of erosions and dilations. Conceptually, however, the direct equation is easier to use and, in this case, probably easier and more efficient to program than either of the compound equations. In turn, combinations of Hit or Miss transformations can be used to construct the morphological operations of sequential thinning.

3.1.5 Sequential Thinning

The Mathematical Morphology operation of sequential thinning can be used to find the 'skeleton' of a set, the 'skiz' of a set, or other transformations depending on the structuring element employed.

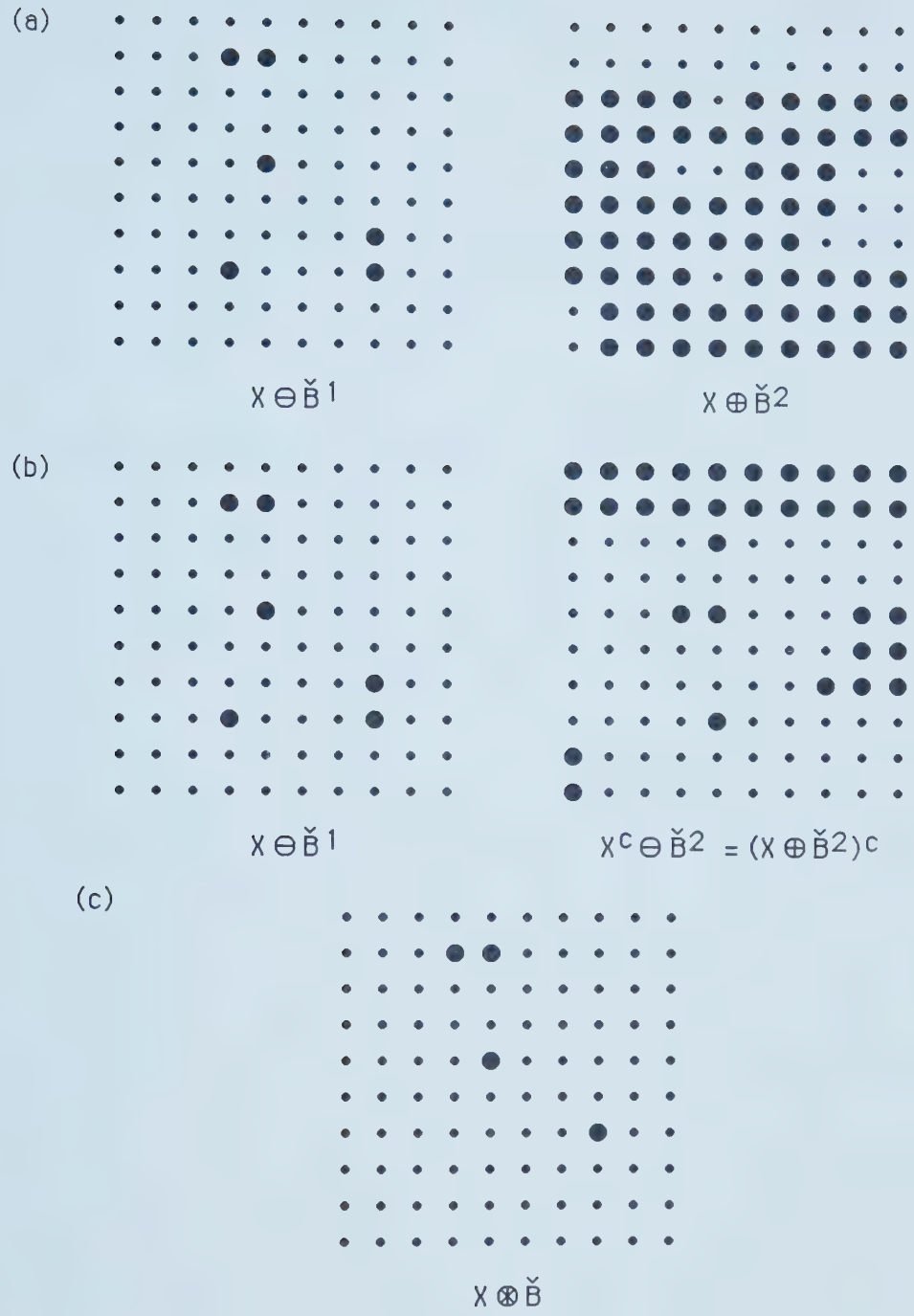


Figure 7. Equivalent methods for computing the Hit or Miss transform in Figure 6. The set difference of the intermediate sets in (a) or the intersection of those in (b) produces (c).

The (non-sequential) thinning of X by B, $(X \circ \check{B})$, results when the Hit or Miss transform is subtracted from X via set differencing:

$$(X \circ \check{B}) = X \mid (X \circledast \check{B}).$$

For example, from Figure 6, $(X \circ \check{B})$ would be the set of points in X less the four points in $(X \circledast \check{B})$. (Thickening is the complement of thinning:

$$(X \odot \check{B}) = X \cup (X \circledast \check{B})$$

but was not found to be of use in our case.)

Sequential thinning involves the use of a sequence of structuring elements {Bi} and repetitions of the sequence until no futher changes occur. The sequential thinning of X by {Bi}, is the successive thinning of X by the series of Bi's:

$$(X \circ \{\check{B}i\}) = (\ldots((X \circ \check{B}1) \circ \check{B}2) \circ \check{B}3 \ldots \circ \check{B}i \ldots).$$

That is, the thinning of X by B1 is then thinned by B2; this result is thinned by B3 and so on.

The sequence {Bi} is usually generated by successive rotations of a basic B. Sequences of two structuring elements for an 8-connected graph are displayed in Figure 8. There are eight different members in each sequence corresponding to the eight possible digital rotations in the common 8-connected square grid; however, the sequences are infinite repetitions of the rotations shown.

For sets, sequential thinning by {Li} gives rise to the 'skeleton' of the image. This is the reduction of the set to a union of single-pixel width arcs which describe the

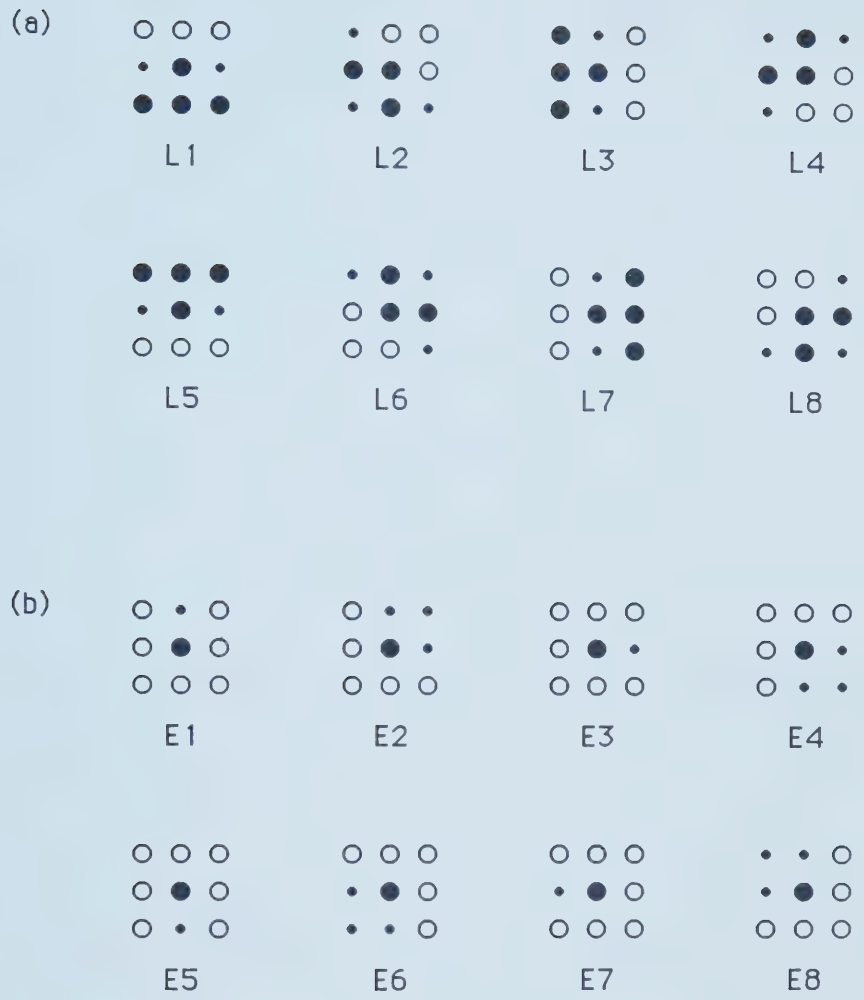


Figure 8. Two structuring elements sequences for sequential thinning: (a) Skeletonization sequence {Li}; (b) Skiz or end-point clipping sequence {Ei}. For each element, the center point is the origin.

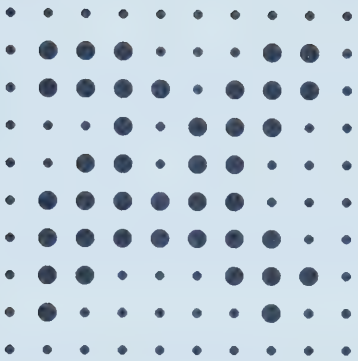
connectivity of the set. Sequential thinning preserves the homotopy of the image. (A transformation is homotopic if it preserves the relative adjacency of connected components and holes; such transformations shrink, expand, distort, or twist but without tearing or merging.)

When (X O {Li}) is then further sequentially thinned by {Ei}, the 'skiz' results. Sequential thinning by {Ei} removes all end-points in the skeleton, leaving only closed loops of single-pixel width. An example of the skeleton and skiz of X are given in Figures 9 (a) and (b), respectively.

Note that after X is successively thinned by some B1,B2,B3,...B8 the successive thinning continues again with B1, then B2, and so on, *ad infinitum*. However, after a finite number of iterations, no further change in the result will occur, and the process can be halted. The successive thinnings can be thought of as removing pixels involved in the configurations exhibited by the sequence of structuring elements. Once all of the (eight) configurations are sequentially removed from the set, any further thinning will be idempotent. This may or may not occur before all the pixels in the set are removed, but effectual termination of the sequential thinning process is guaranteed.

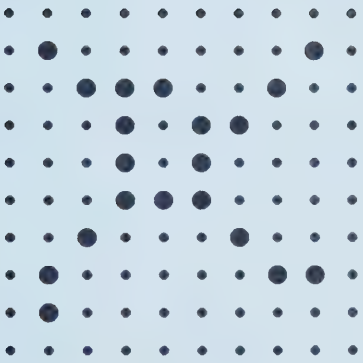
3.1.6 Extension from Sets to Gray-Tone Images

The transformations of erosion and dilation, and thus opening, closing, the Hit or Miss transform, and thinning can be performed on gray-level images X by dividing these



the set X, ●

(a)



(b)

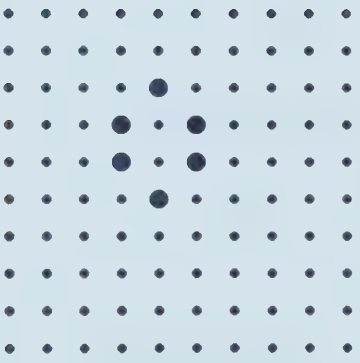


Figure 9. The skeleton (a) and the skiz (b) resulting when X is sequentially thinned by {Li} and then further sequentially thinned by {Ei}.

images into a family of component sets or levels. These levels, X_t, are constructed by considering those pixels from the image with values greater than or equal to a threshold t:

$$X_t = \{(x,y): X(x,y) \geq t\}.$$

In Figure 10(a) a very simple image with only four (including zero) gray-levels is decomposed into its three composite levels. Always, $X_{t+1} \subset X_t$, as can be seen from the figure.

To dilate (erode) image X, a dilation (erosion) is performed on each level X_t separately, and the result from each level X_t is that level Y_t of the resulting image Y:

$$Y_t = (X_t \oplus \check{B}) \qquad (Y_t = (X_t \ominus \check{B})).$$

This maintains the fact that $Y_{t+1} \subset Y_t$. The levels of the dilation and erosion of X by B from Figure 10(a) are shown in Figures 10(b) and (c), respectively. The resultant image Y is simply that image composed of the various levels Y_t.

A more efficient definition of the dilation (erosion) of a gray-tone image is the following:

$$Y(x,y) = (X \oplus \check{B})(x,y) = \max\{X(n,m);\ (n,m) \in B_{xy}\}$$

$$(Y(x,y) = (X \ominus \check{B})(x,y) = \min\{X(n,m);\ (n,m) \in B_{xy}\})$$

where Y(x,y) denotes the value at pixel (x,y) of image Y. In words, the dilation (erosion) of the image X by B is performed by moving the structuring element B over X and, at each pixel, writing to the corresponding pixel in Y the maximum (minimum) of all the pixels overlayed by the B. Thus, the dilated (eroded) image can be computed pixel by

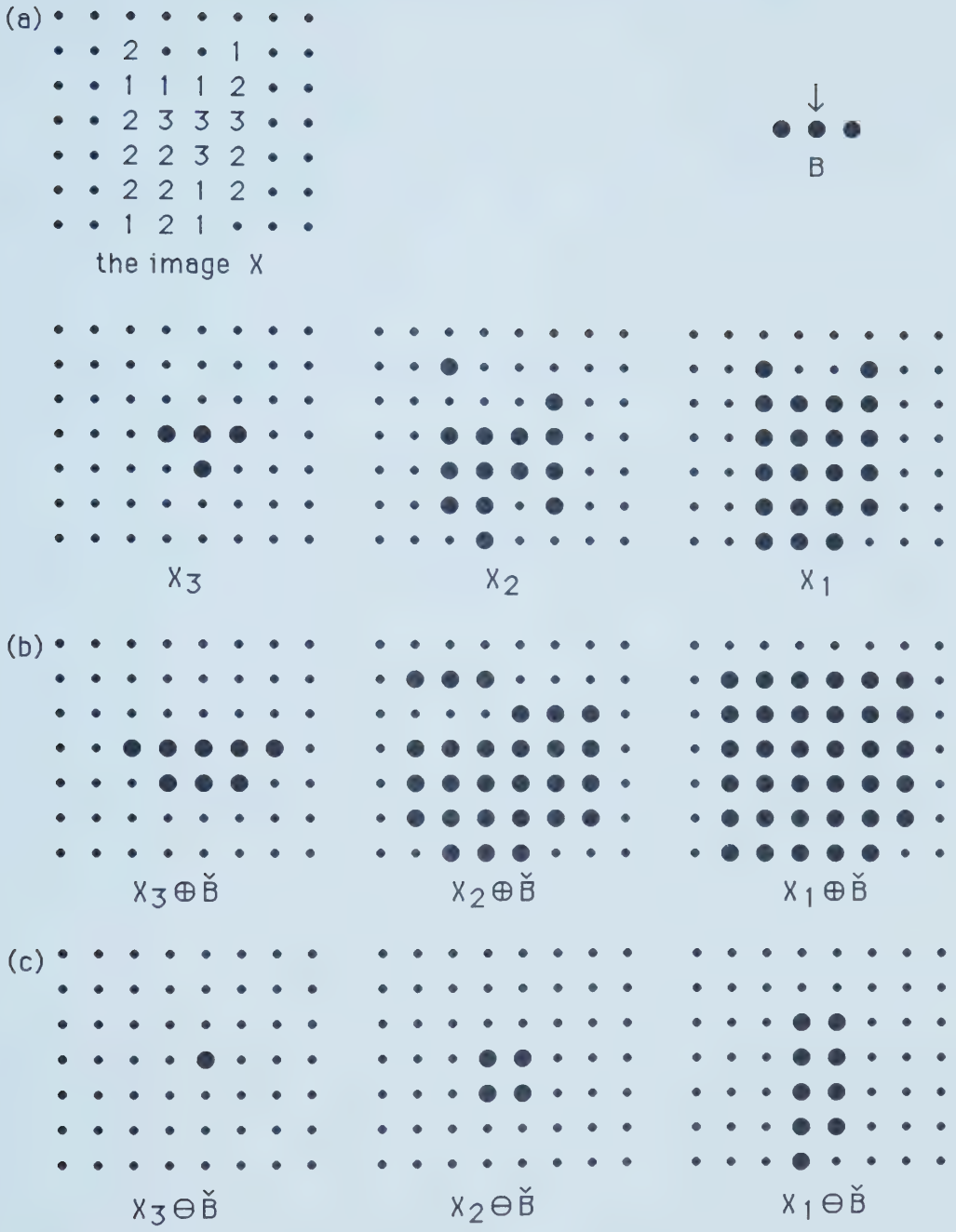


Figure 10. (a) A simple grey-tone image X decomposed into its composite levels X_t; (b) The levels of the dilation $(X \oplus \check{B})$; (c) The levels of the erosion $(X \ominus \check{B})$.

pixel without resorting to level decomposition and recombination.

To open (close) the input image X, each level of X may be opened (closed) and the resulting levels recombined to obtain the opening X_B (closing X^B). Alternatively, the opening (closing) can be obtained by using the efficient definition of the erosion (dilation) to produce an intermediate image Y' on which a similar dilation (erosion) is performed.

The Hit or Miss transformation, and thus thinning, applied to a gray-tone image is not so straight forward. The Hit or Miss transform or thinning of a level X_t is not that level Y_t of the resultant Y since set Y_{t+1} is not necessarily included in the set Y_t. An example of thinning X by L is given in Figure 11. Figure 11(a) shows the three levels X_t of image X. Thinning each level by L would remove (zero) the circled points. Notice, however, that:

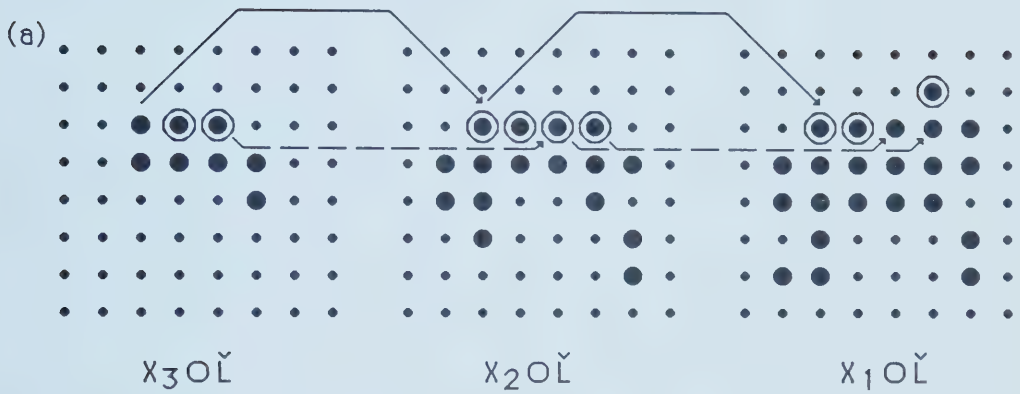
$$(X_2 \circ \check{L}) \subset (X_1 \circ \check{L})$$

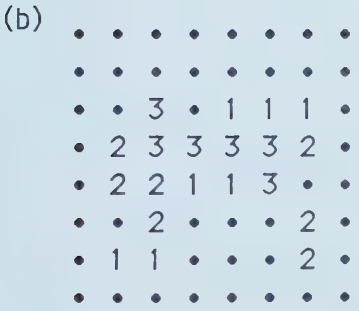
$$\text{but } (X_3 \circ \check{L}) \not\subset (X_2 \circ \check{L})$$

$$\text{and } (X_3 \circ \check{L}) \not\subset (X_1 \circ \check{L})$$

as the solid arrows indicate. This suggests that the pixel associated with the solid arrows should remain in the result as a pixel of value three. The pixels removed by thinning levels three and two associated with the dashed arrows are not removed by thinning level one. As with erosion, these pixels remain in the output image but with their value reduced to the level in which they survive (level one in

the image X





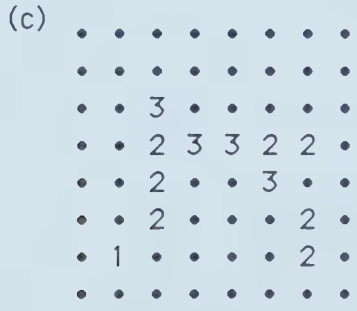


Figure 11. The thinning of gray-tone image X by the structuring element L and the sequence {Li}: (a) The three levels X_t thinned by L; (b) The lower thinning (X O Ľ); (c) The lower sequential thinning (X O {Ľi}).

this case). The final result of X thinned by L appears in Figure 11(b). Only those pixels completely removed from all levels of X are excluded (zeroed) in the image $(X \circ \check{L})$.

The type of thinning a gray-tone image exemplified in Figure 11 is called 'lower thinning' since it starts from the top levels and works downward. The levels Y_t of the lower thinned image $Y = (X \circ \check{B})$ are defined by

$$Y_t = \bigcup_{i=t}^{m} (X_i \circ \check{B})$$

where m is the maximum level of the image X:

$$1 \leq i \leq m.$$

Thus, pixels removed from a level t by thinning will be restored to that level by the union if they were not removed from a higher level $i > t$.

An equivalent but more efficient definition for the lower thinning is the following using the two parts B^1 (•) and B^2 (o) of a structuring element B:

$$\text{If for all } (n,m) \in B^1_{xy}, \quad X(n,m) \geq X(x,y)$$

$$\text{and for all } (k,l) \in B^2_{xy}, \quad X(k,l) < X(x,y)$$

$$\text{then } Y(x,y) = \max\{X(k,l): (k,l) \in B^2_{x,y}\},$$

$$\text{otherwise } Y(x,y) = X(x,y).$$

This says, to compute a pixel (x,y) of the lower thinned image Y, the structuring element B is centered at the corresponding pixel in X and a test is made for whether all the pixels overlayed by B^1 in X are greater than or equal to pixel (x,y) in X and, in addition, all the pixels overlayed by B^2 in X are less than pixel (x,y). If the test is true, we assign the maximum of the values of those pixels in X

overlayed by B^2 to pixel (x,y) in Y; otherwise, if the test is false, we simply copy the value of pixel (x,y) in X to Y.

Lower sequential thinning of a gray-tone image is performed simply by lower thinning each resulting image, in turn, by the next structuring element in the sequence. If the {Li} sequence is used, the lower skeleton results, and if the lower skeleton is further sequentially thinned by the sequence {Ei}, then the lower skiz or lower clipped skeleton results.

Lower thinning by {Li} emphasizes 'ridges' within the image by reducing these ridges to a single-pixel width while suppressing the background. Lower skeletonization does not retain ridges exactly since portions of a ridge may be reduced in value. Even so, the ridge will usually stand significantly above the background. To see this, consider Figure 11(c) which shows the lower sequential thinning of X by the sequence {Li}.

Illustrations of the transformations incurred during lower skeletonization are shown in Figure 12 where a 1-D image is displayed in profile or side view, height indicating gray-level. The image is to be sequentially lower thinned by the sequence {L'} consisting of only repetitions of the single modified structuring element L'. Suppose L' is applied to the second pixel in X as shown in Figure 12(a). Using the efficient definition of the lower thinning equation, since ($c \geq b$) and ($a < b$), pixel b in the output is given the maximum of a, as shown in Figures 12(a) and (b).

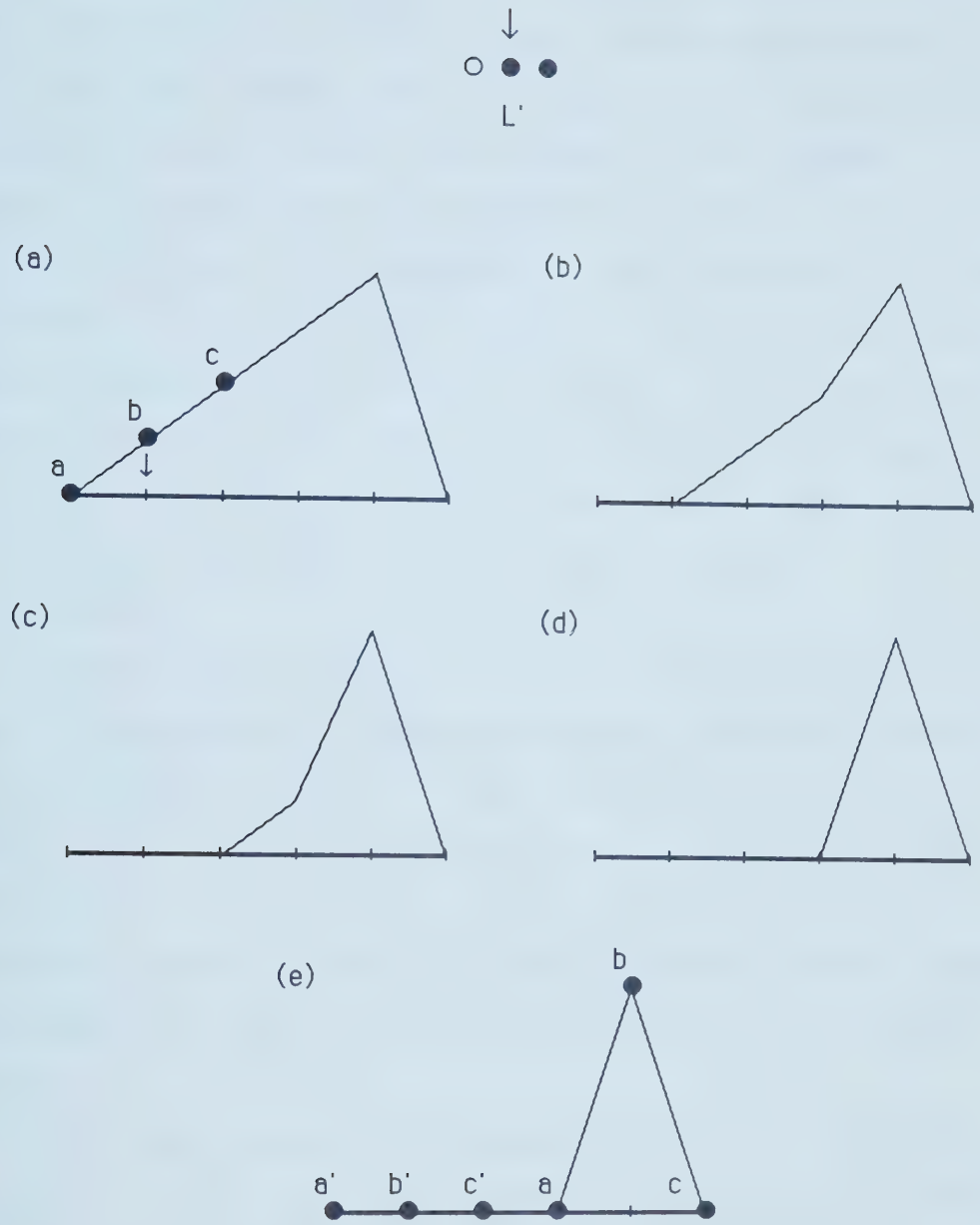


Figure 12. A 1-D example of the transformations incurred during lower skeletonization (X O{L'}): (a) The image is shown in profile; (b) After the first application of L'; (c),(d) After successive lower thinnings; (e) No further change will occur.

The structuring element is then shifted by one pixel and the operation repeated. After the first complete application of L', the lower thinned image in Figure 12(b) results. In this example, the next structuring element in the sequence is again L'; so we lower-thin the previous result again by L' obtaining Figure 12(c). Subsequent lower thinning gives rise to Figure 12(d). No further change in the image will result from successive thinnings by this sequence {L'} since, as shown in Figure 12(e), there is no case where ($c \geq b$) and ($a < b$). In the 2-D case, the different rotations of L in {Li} reduce, to single-pixel width, ridges of various orientations.

Further lower sequential thinning of the lower skeleton by {Ei} suppresses all end-points of the emphasized ridges. Any non-closed, dendritic ridges found by lower skeletonization are de-emphasized and brought down to the background level. A simple example of the lower clipped skeleton appears in Figure 13. As illustrated, lower clipped skeletonization maintains only closed, single-pixel width ridges.

With the Mathematical Morphology transformations of opening, closing, and lower sequential thinning by {Li} and {Ei} on gray-tone images, all the tools necessary for developing an automated contour detection algorithm for CT scans of the wrist are present.

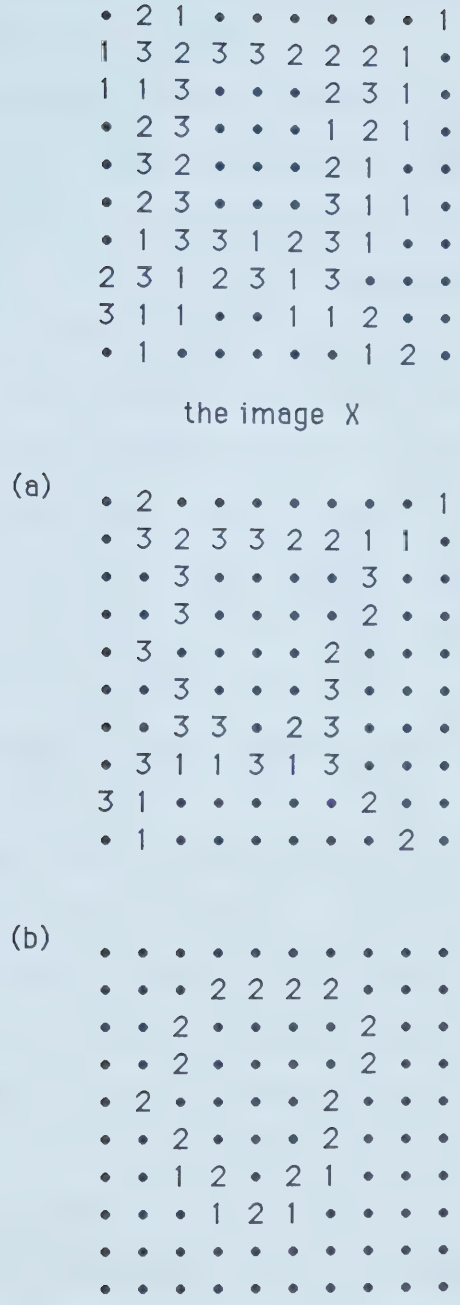


Figure 13. (a) Lower sequential thinning of X by: {Li} producing the lower skeleton; then (b) further lower sequential thinning by {Ei} producing the lower clipped skeleton.

3.1.7 Contour Detection by the Algorithm of Lower Clipped Skeletonization (LCSKEL)

Lower sequential thinning by {Li} emphasizes ridges within a gray-level image by reducing these ridges to a single-pixel width while suppressing the background. Further lower sequential thinning by {Ei} suppresses any end-points of these ridges, maintaining only closed single-pixel width ridges. Since the CT bone images that were investigated consist of a dense cortical shell, filled with less dense trabecular bone and surrounded by soft-tissue of much lower density, lower clipped skeletonization should highlight the cortical shell, reducing it from within and from without to a single-pixel width closed ridge of relatively strong intensity. Thresholding may then be used to extract this main cortical ridge from the background. The trabecular bone and, to a lesser extent, the surrounding soft-tissue may give rise to some closed ridges of their own, but these will generally be of much lower intensity and should be excluded by the thresholding. Even if closed trabecular ridges are relatively strong, they will appear interior to the main cortical ridge and will not affect the cortical shell determination by LCSKEL.

A flowchart representation of the LCSKEL algorithm is shown in Figure 14. The CT image is input (usually with the ulna already 'erased' using a manual trackball and cursor method), and any negative values are zeroed. This zeroing is simply for convenience and consistency with other processing

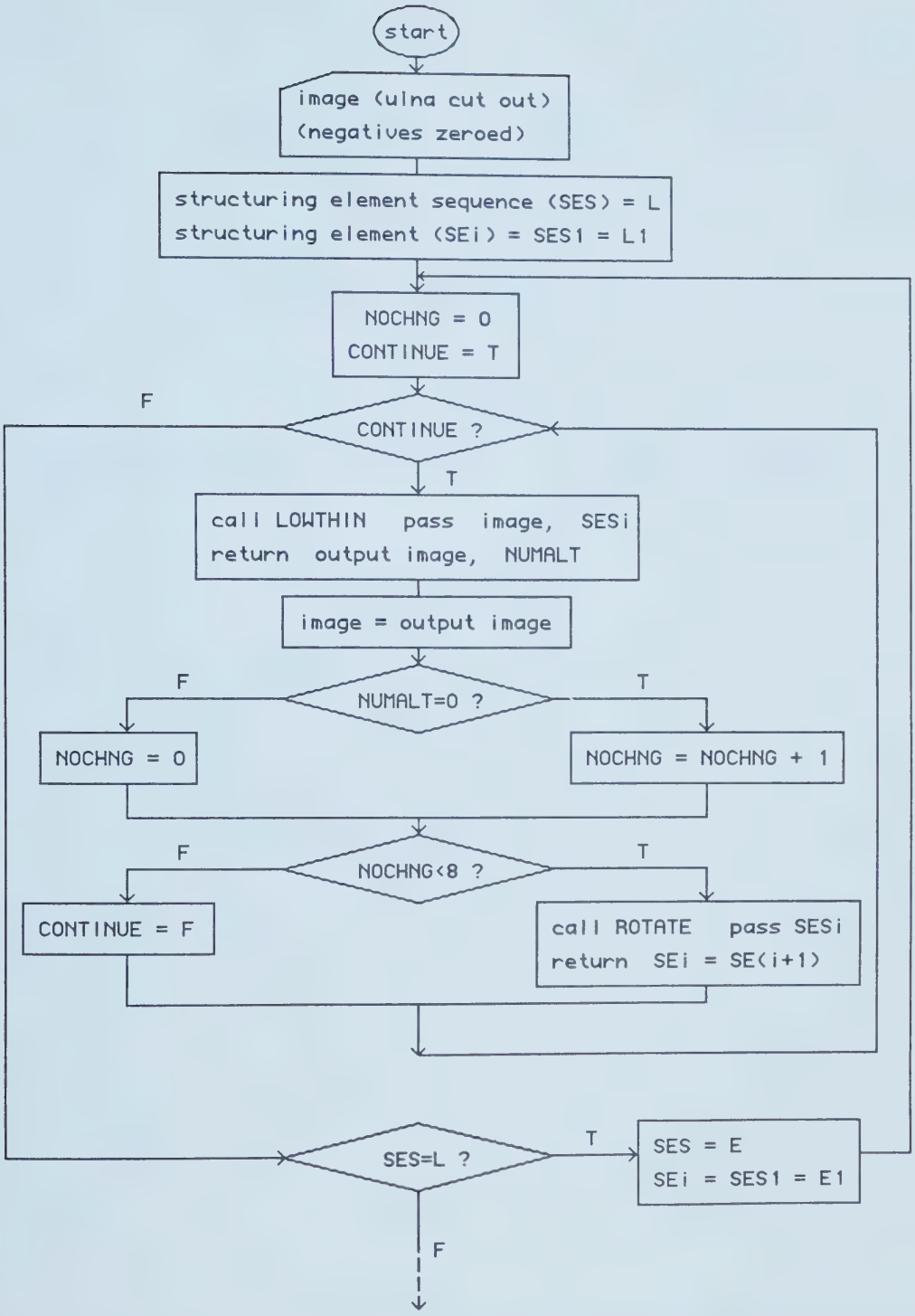


Figure 14. Algorithm flowchart for contour detection by lower clipped skeletonization.

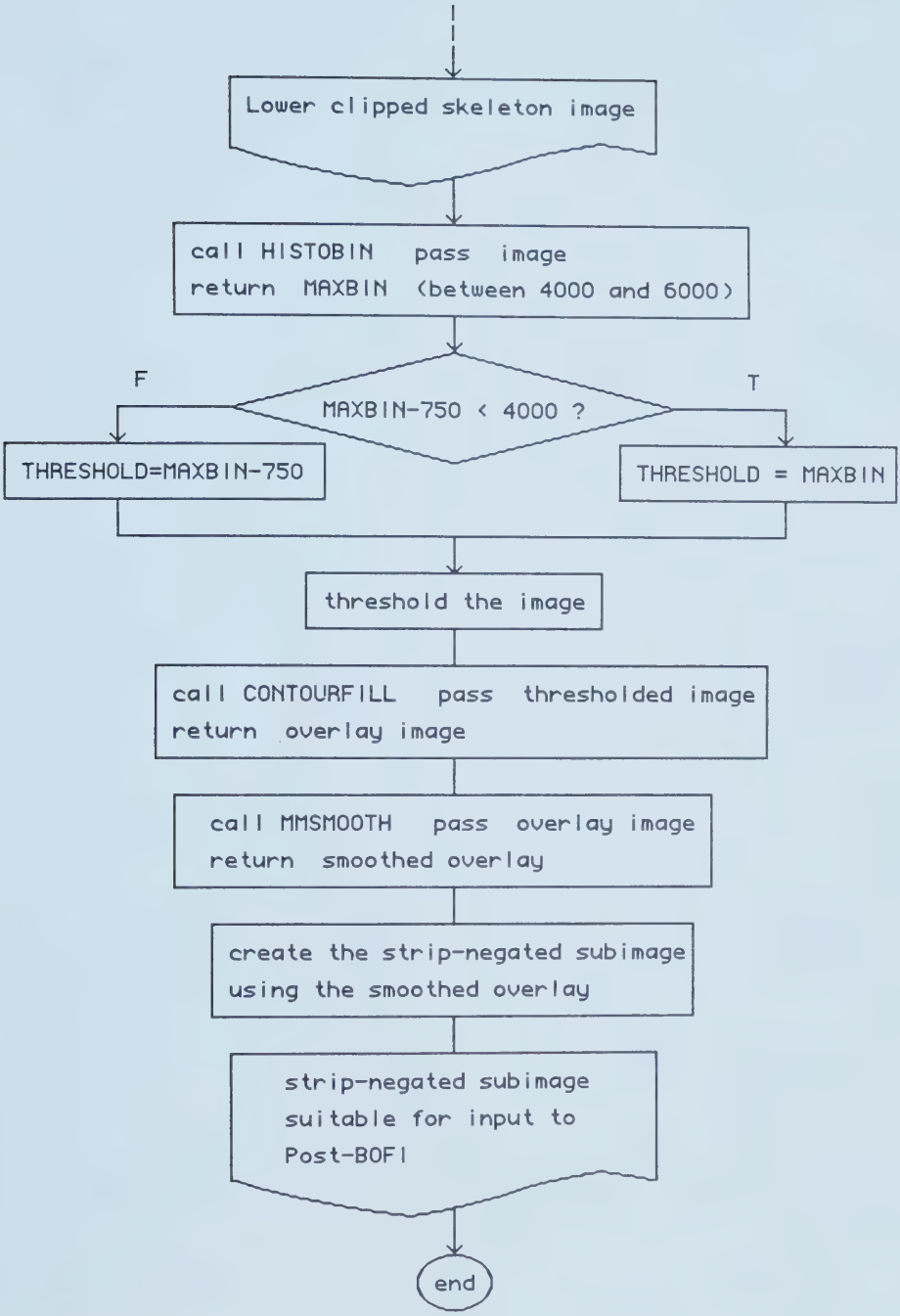


Figure 14 continued.

and is not otherwise necessary. (Negative values are a result of the mathematical reconstruction process and are physically meaningless.) An example of such an input CT image is shown in Figure 15(a) using a color map. The 0 to 255 gray-level display is divided equally into the sixteen color bands shown.

The {Li} structuring element sequence is first chosen (SES=L) and the first structuring element is initially set to L1 (SEi=SES1=L1). NOCHNG is a count of the number of times lower thinning is performed without altering the input image; it is initially set to zero. CONTINUE is set to true until there are eight consecutive lower thinnings performed with no change to the image (NOCHNG=8). While CONTINUE is True, subroutine LOWTHIN is called to lower-thin the gray-tone image using structuring element SEi. LOWTHIN returns both the lower thinned resultant image and a count of the pixels that were altered with respect to the input image (NUMALT). Of course, if NUMALT=0, then NOCHNG is incremented. Whenever NUMALT≠0, NOCHNG is reset to zero. As long as the image is not left unaltered for eight consecutive passes (NOCHNG<8), then lower thinning will be sequentially repeated using the next (rotated) structuring element in the sequence (SEi=SES(i+1 mod 8)). Finally, when NOCHNG≥8, CONTINUE is set to False since the lower sequential thinning with {Li} is complete.

All the ridges in the original image are now of single-pixel width and, especially, the main cortical ridge

Figure 15. (a) A typical input CT image for LCSKEL. The ulna has been manually removed by tracing a separation line with the cursor and trackball system. (b) The lower skeleton of (a). (c) The lower clipped skeleton of (a) produced by end-point clipping (b).

stands out above the background as seen in Figure 15(b) which displays the results of lower thinning Figure 15(a). However, many dentritic ridges protrude from the cortical ridge, and these must be removed.

Now the structuring element that was SES=L is changed to SES=E, and the first structuring element is initialized (SEi=SES1=E1). NOCHNG is reset to zero and CONTINUE to True so that the lower sequential thinning process can be repeated but now using {Ei} on the lower skeleton. When NOCHNG reaches 8, we again exit the CONTINUE loop.

The resulting image is now the lower clipped skeleton, and only closed ridges remain. The lower clipped skeleton from Figure 15(a), produced by end-point clipping Figure 15(b), is shown in Figure 15(c). The most obvious closed ridge is the main cortical ridge, but other ridges in the soft-tissue and trabecular regions have survived. These are generally of much lower intensity than the cortical ridge; so they can be removed by thresholding.

The lower clipped skeleton image is analyzed to find the most frequently occuring pixel value (MAXBIN) between 4000 and 6000, 3500 to 5000 being the ideal value for cortical bone in our images. Pixels of value MAXBIN will almost certainly belong to the main cortical ridge. THRESHOLD is set to a value 750 less than MAXBIN; however, if this brings THRESHOLD below 4000, then 750 is not subtracted leaving THRESHOLD=MAXBIN. The image is then thresholded; any pixels less than THRESHOLD are zeroed.

Choosing THRESHOLD less than MAXBIN ensures that all of the the cortical ridge is preserved. A much lower value would allow unwanted closed ridges adjacent to the cortical ridge to survive.

There generally existed a wide latitude in selecting an acceptable threshold for the types of images that were analyzed: from about 200 to about 1200 below MAXBIN. The chosen value of 750 below MAXBIN was close to the middle of this range and was found to give the best overall results.

The thresholded image contains the cortical ridge contour and often some interior closed ridges due to the trabecular bone being of sufficiently high density to exceed the threshold (Figure 16(a)). These interior ridges are of no consequence since the outermost contour is filled. In the filling process, all contour and interior pixels are given equal value creating essentially a binary image which is the radius bone overlay as shown in Figure 16(b).

This overlay may have slight protrusions or inclusions of a few pixels in size; so a small amount of smoothing is performed. Subroutine MMSMOOTH uses an opening by an octagon with sides of length 3 followed by a closing using a square of length 7. The opening, which keeps only those pixels into which the octagon fits, shaves off any protrusions of size 3 or smaller. The closing then fills in any inclusions across which the square can span; that is, inclusions of 5 or fewer pixels wide. For the example in Figure 16(b), smoothing does not result in any significant changes to the overlay.

Figure 16. (a) The lower clipped skeleton from Figure 15(c) after thresholding (at MAXBIN-750 = 5227) and (b) after the resulting contour is filled to create the bone overlay.

This smoothed radius overlay is then used to create a strip-negated subimage from the original CT image which is compatible for viewing and further processing by the current Post-BOFI procedure. This 128x128 subimage is automatically centered about the center of intensity of the bone overlay so that it frames the radius. The appropriate pixels from the original non-negative CT image are copied to this subimage. Those pixels which are outside the pixels corresponding to the bone overlay are negated in order to effectively isolate the radius (from the cortical ridge inward) from the rest of the CT image.

With output of this strip-negated subimage, the LCSKEL algorithm ends.

3.2 Cross-Correlation

As its name implies, EDCORP uses cross-correlation to detect bone contours. Cross-correlation is a common operation in digital signal processing [6,29,30] and is usually performed between two signals (auto-correlation is computed between a signal and itself). However, cross-correlation, or match filtering as it is sometimes called, may be performed between a signal and a kernel or filter (which may be considered a short signal). The objective of cross-correlation is to determine the amount of shift between the two signals that is required to bring them into best correspondence or similarity and to give a measure of this correspondence. Effectively, one signal is shifted

with respect to the other, and a measure of their match is computed for each relative shift, producing a correlation function.

It was desired to determine the best location of match between a profile through the cortical bone shell (the signal) and an idealized or model bone profile for which the exact location of the bone edge has been determined (the correlation kernel). By choosing appropriate profiles through the cortical bone and determining by cross-correlation the 'best' locations of the bone edge on these profiles, a complete bone edge contour may be produced.

3.2.1 Profiles for Cross-Correlation

To determine the location of the bone edge, profiles or cross-sections through the cortical bone are employed. Ideally, these profiles should be orientated perpendicular to the cortical bone so that the transition from bone to soft-tissue is most abrupt. An example of the orientation of profiles for a model radius is shown in Figure 17(a). Since this ideal radius is circular, the profiles emanate from an origin chosen at the center of the bone. Figure 17(b) displays one of these ideal profiles. It is simply a cross-section through the trabecular bone, across the cortical shell, and out into the soft tissue. The outer edge of the cortical bone is obvious.

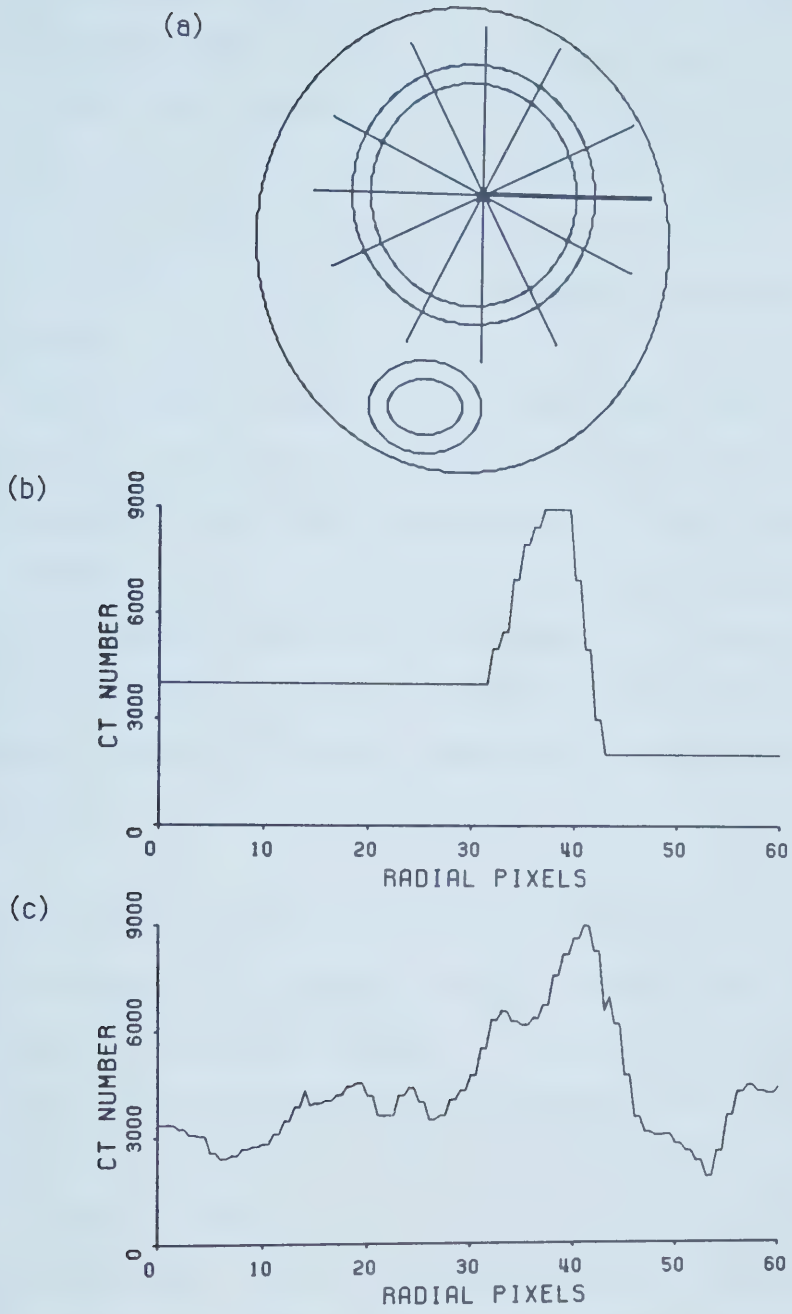


Figure 17. Example of profiles through a model radius: (a) Idealized model of a radius bone showing the orientation of profiles; (b) One of these radial profiles showing the location of the bone edge; (c) Radial profile with the effects of measurement noise, reconstruction error, and blurring considered.

In the real world situation, this edge is not so obvious. Measurement noise, image reconstruction error, and finite beam width result in a noisy and blurred CT image and, therefore, a less than ideal profile as illustrated in Figure 17(c). The outer edge of the cortical bone is usually taken as the mid-level of the ramp edge that passes through the average of the noise.

Real radius bones are not circular but they are closed and usually elliptical in shape. Profiles radially emanating from a common origin near the center of such a bone would not everywhere be perpendicular to the cortical shell. As a profile becomes oriented more obliquely to the cortical ridge, a broadening of the ramp edge occurs. The more circular the radius however, the more this broadening is minimized.

3.2.2 Cross-Correlation and the Correlation Function

The result of cross-correlation between two signals or between a profile and a kernel is the cross-correlation function. The cross-correlation function C(r) is a measure of the similarity between the kernel q(k) and the profile p(k) at each relative shift [29,30]. As the kernel slides over the profile, their corresponding values are multiplied and the products summed giving the value of the correlation at each point:

$$C(r) = \sum_{k} p(k)q(k-r).$$

At the relative shift 'r' for the best match between the

kernel and the profile, the correlation function takes on its maximum value.

If the kernel represents a region about a bone edge and the exact location of the bone edge is marked by placing this point at the origin of the kernel, then upon cross-correlation of this kernel with a profile, the 'best' location of the bone edge in the profile will correspond to the location of the maximum value in the correlation function. At the relative shift indicated by this location, the kernel best matches the profile and the edge point (origin) in the kernel is taken as the edge point in the profile. This is illustrated in Figure 18 where the profile from Figure 17(c) is cross-correlated with a kernel to compute the correlation function.

Cross-correlation is identical to linear convolution except that the kernel or filter is not transposed with respect to its origin [29]. Thus, the correlation function can be considered a filtered version of the input profile where the filter is the transpose of the correlation kernel. The locating of the bone edge point at the kernel's origin simply induces a shift in the filtered output.

Note that some texts define cross-correlation as a normalized function [6,29,31] and that these definitions do not always agree. Normalization ensures that $0 \leq C(r) \leq 1$, $C(r)=1$ being a perfect match and occuring when $p = cq$ for some relative shift 'r', c being a multiplicative constant.

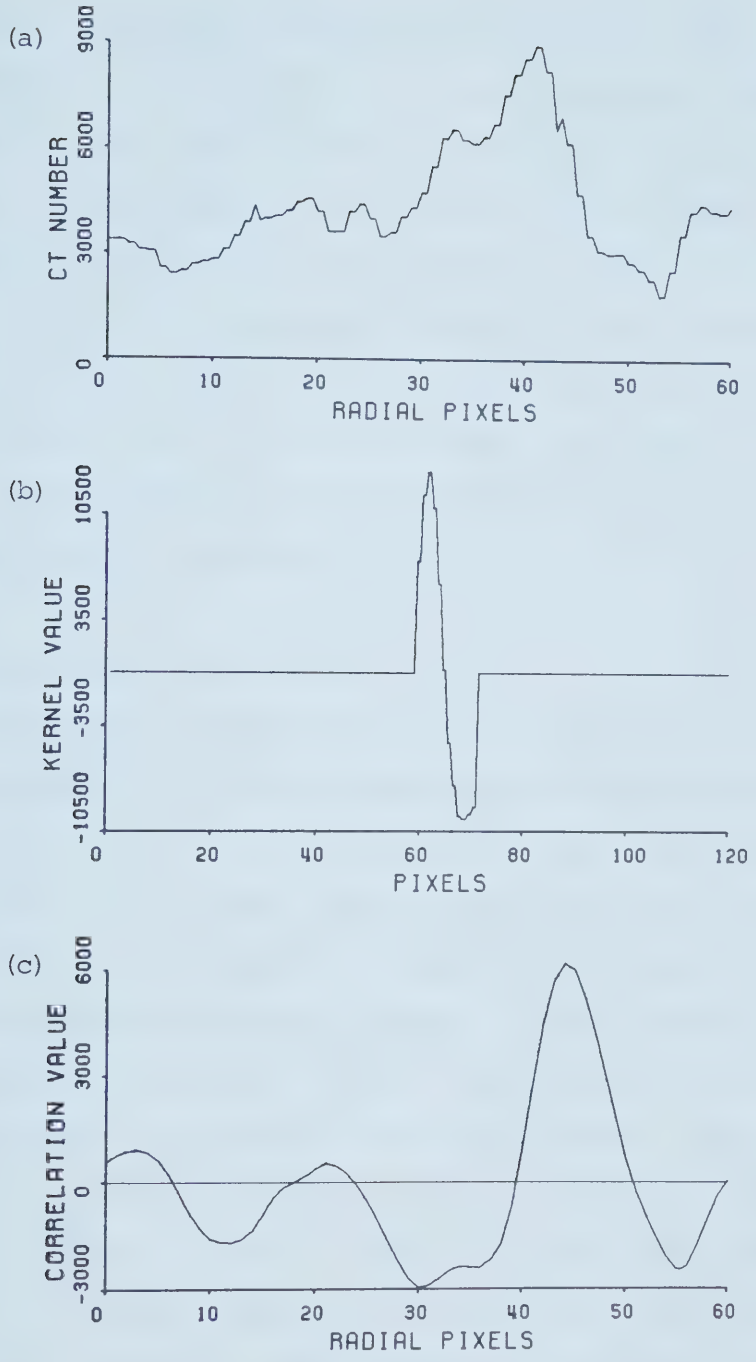


Figure 18. Example of a cross-correlation function: (a) The radial profile from Figure 17(c); (b) A correlation kernel; (c) The correlation function resulting from cross-correlating (a) and (b).

Normalization may depend on the geometric mean of the 'energies' of the signals or on a local weighted average of the profile (the kernel giving the weights). For our purposes, normalization is unnecessary since we are not interested in the value of the correlation but only in the location of the local maximum. We do not want the local average level of the profile to influence the match with the kernel at that locality. That is, only the shapes of the kernel and the profile are to be matched; their relative strengths are not important to us.

3.2.3 The Correlation Kernal

Since cross-correlation is being used to try to locate the bone edge in profiles exhibiting noise and blurring, the correlation kernel should simulate an idealized profile of such an edge. The kernel employed was empirically determined from a CT scan of a phantom composed of an aluminum ring (with X-ray attenuation very close to compact bone) surrounded by plexiglass (simulating soft-tissue). Though the noise from one profile to the next varies, the effects of a finite beam width remain relaively constant. By creating the kernel from an actual CT image, this beam width effect is included in the kernel and allows for a better match to all profiles. The effects of the slight amount of noise in this kernel are assumed to average out in the cross-correlations process.

The profile through the CT phantom is displayed in Figure 19(a). The portion shown in bold was used for the kernel. This portion encompasses the character of the type of bone edges that are sought, containing both the cortical peak and the drop to soft-tissue. It was found that as long as the general character of the correlation kernel was not altered, results were not significantly dependent upon slope, amplitude, or size. This also suggests that a single kernel can be used for cross-correlation between quite different profiles with consistently good results.

For proper cross-correlation, the kernel must have zero mean value, thus the selected portion has been shifted down by its original mean as shown in Figure 19(b). This zero mean criterion is not clearly stated in standard texts for signal processing. Consider a kernel $Q(k) = q1(k) + q2(k)$ as composed of that component $q1(k)$ shown in Figure 18(b) plus a DC component $q2(k)$ with amplitude equal to the kernel's original mean value and a width equal to that of the kernel as shown in Figure 19(c). Since cross-correlation is a linear operation, cross-correlation with $Q(k)$ equals adding the results of cross-correlation with both components:

$$C(r) = \sum_k p(k)\{q1(k-r) + q2(k-r)\}$$
$$= \sum_k p(k)q1(k-r) + \sum_k p(k)q2(k-r).$$

Cross-correlation of a signal by the symmetrical box $q2(k)$ is essentially an average filtering:

$$\sum_k p(k)q2(k-r) = \sum_k p(k)q2(r-k)$$

since $q2(k-r) = q2(r-k)$. When this result is added to the

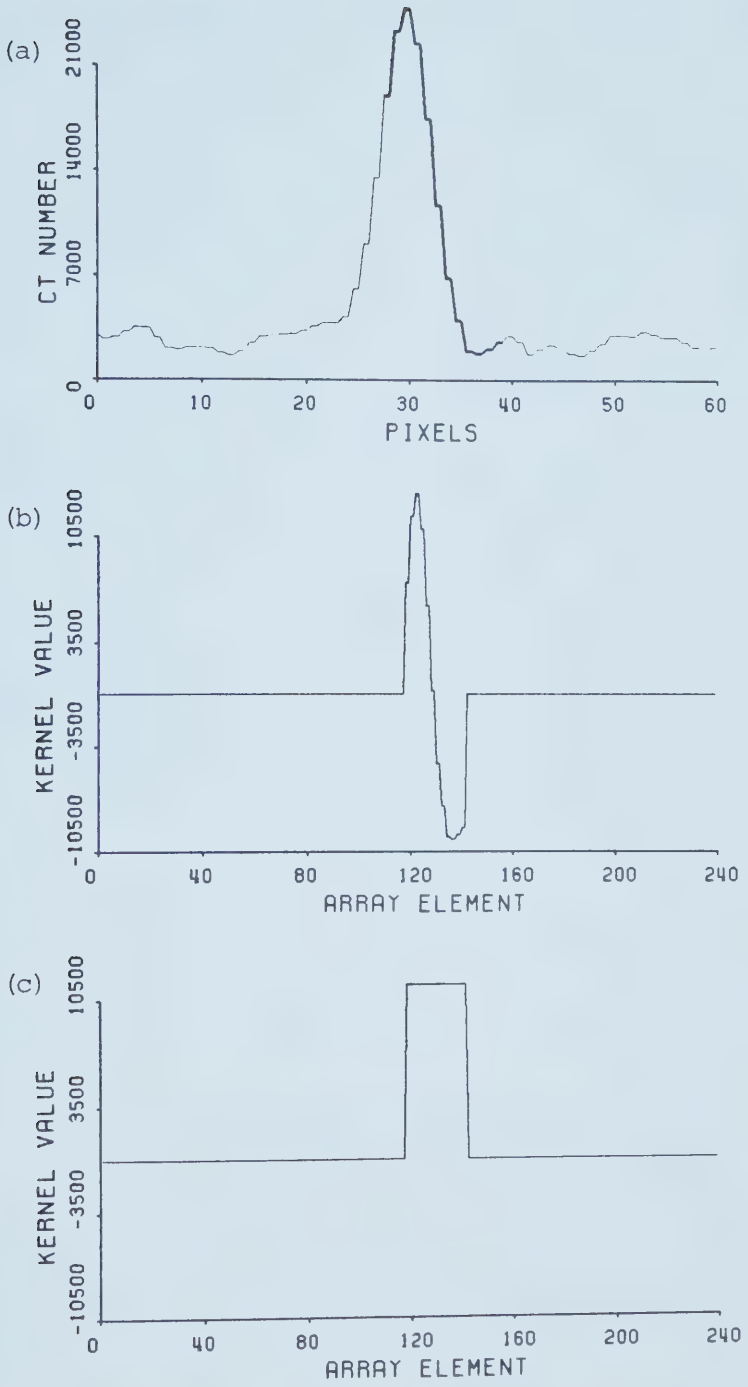


Figure 19. Empirical construction of the cross-correlation kernel: (a) Profile through a phantom consisting of an aluminum ring in plexiglass; (b) The correlation kernel; (c) The DC component which was removed in producing (b) from (a).

cross-correlation by q1(k), the cross-correlation function will be increased by an amount that depends upon the local average of the signal being cross-correlated. For example, in our case, the peak of the cross-correlation function will be shifted towards the location of the maximum in the profile (to the left). This effect is clearly not desirable since it will skew our determination of the edge contour toward the interior of the radius.

The profile need not additionally have a zero mean when a short correlation kernel is being used. Consider the finite profile p(k) as now composed of a zero mean profile component p1(k) plus a box component p2(k) of identical length and an amplitude equal to the original mean value v:

$$p(k) = p1(k) + p2(k) \quad \text{for } k1 \le k \le k2$$
$$= 0 \quad \text{otherwise}$$

where

$$p2(k) = v \quad \text{for } k1 \le k \le k2$$
$$= 0 \text{ otherwise.}$$

Then

$$C(r) = \sum_k \{p1(k) + p2(k)\}q(k-r)$$
$$= \sum_k p1(k)q(k-r) + \sum_k p2(k)q(k-r).$$

If the kernel has zero mean value, cross-correlation C2(r) with the box p2(k) will give zero everywhere within the bounds of the box less the kernel size:

$$C2(r) = \sum_k p2(k)q(k-r)$$
$$= v \sum_{k=k_1}^{k_2} q(k-r)$$
$$= 0 \quad \text{for } (k1-n1) \le r \le (k2-n2)$$

since

$$\sum_{k=k_1}^{k_2} q(k-r) = \sum_{m=k_1-r}^{k_2-r} q(m)$$

$$\text{and } \sum_{m} q(m) = 0,$$

where n1≤n≤n2 is the extent of the kernel. In this region then, cross-correlation is not dependent upon the mean value of the profile.

In addition to being shifted downward so as to have zero mean value, the kernel is also shifted laterally so that the exact location of the edge is at the origin. This step conveniently ensures that upon cross-correlation, the location of the maximum value in the correlation function corresponds to the best location of the edge in the profile. Notice that the accuracy of the edge point determined via this cross-correlation may be adjusted by shifting the kernel in its array. If the kernel is shifted to the right, for example, the location of the maximum value in the correlation function will be shifted to the left.

For programming convenience, the origin of the kernel is considered as array element 128 in order to avoid negative array elements. To account for this origin relocation, a computational shift of the kernel to the left by 128 is required in the cross-correlation operation:

$$C(r) = \sum_{k} p(k)q(k+128-r).$$

The properly positioned kernel is as shown in Figure 19(b).

3.2.4 The Algorithm of Edge Detection by Cross-Correlation of Radial Profiles

EDCORP uses profiles emanating radially from an origin automatically chosen at the center of the radius bone. These profiles are cross-correlated with the kernel developed in the previous section. The location of the maximum value in the correlation function indicates the relative shift between the kernel and the profile for best match and, thus, also indicates the location of the bone edge in the profile. By using radial profiles at sufficiently close angular increments to adequately sample the cortical shell and then, when necessary, connecting the so determined bone edge points, a complete contour of the bone can be tracked.

Figure 20 shows a flowchart representation of the EDCORP algorithm. The CT image is input (usually with the ulna left in), and any negative values are zeroed. An example of an input CT image is displayed in Figure 21(a).

To automatically determine the origin from which the radial profiles will emanate, the center of intensity of only those pixels over a threshold of 4000 is calculated. This value of 4000 was chosen to exclude most of the image except for the cortical bone. Since the ulna is present, this center of intensity will be shifted away from the true center of the radius, towards the ulna. When the ulna is very far from the radius and of comparable intensity and size, as in the one-third distal scans (scans taken through the midshaft), this center of intensity will fall roughly

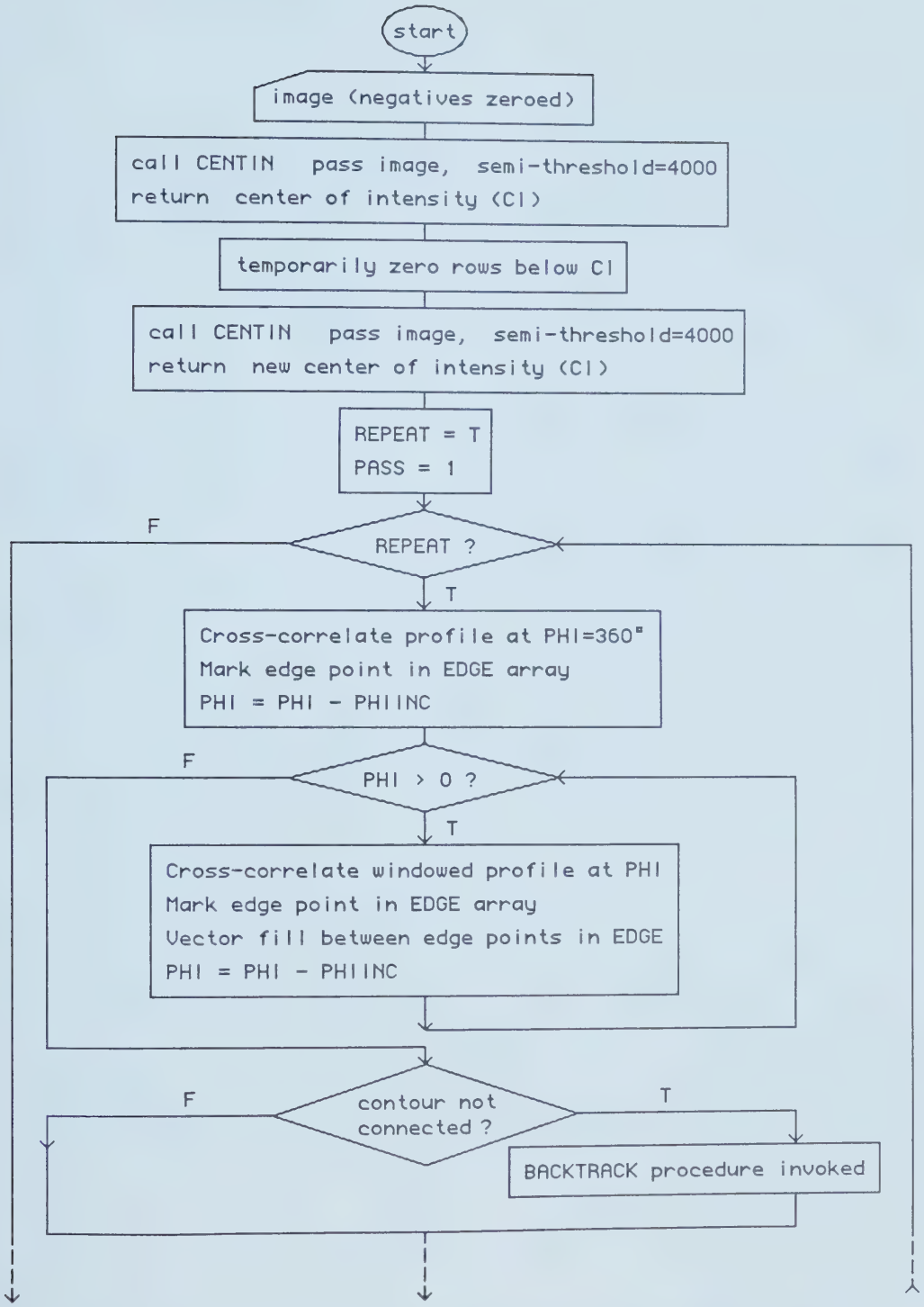


Figure 20. Algorithm flowchart for edge detection by cross-correlation of radial profiles.

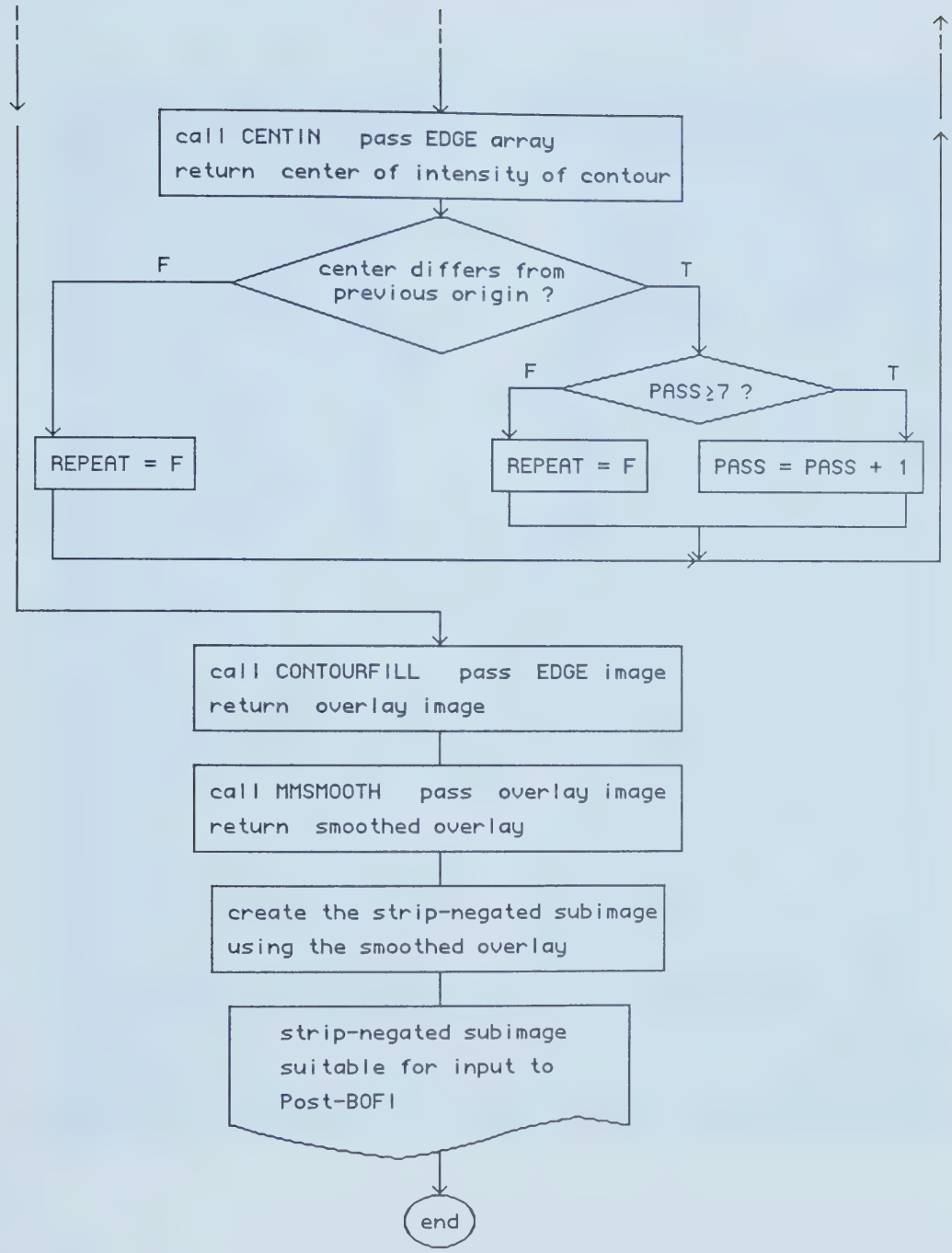


Figure 20 continued.

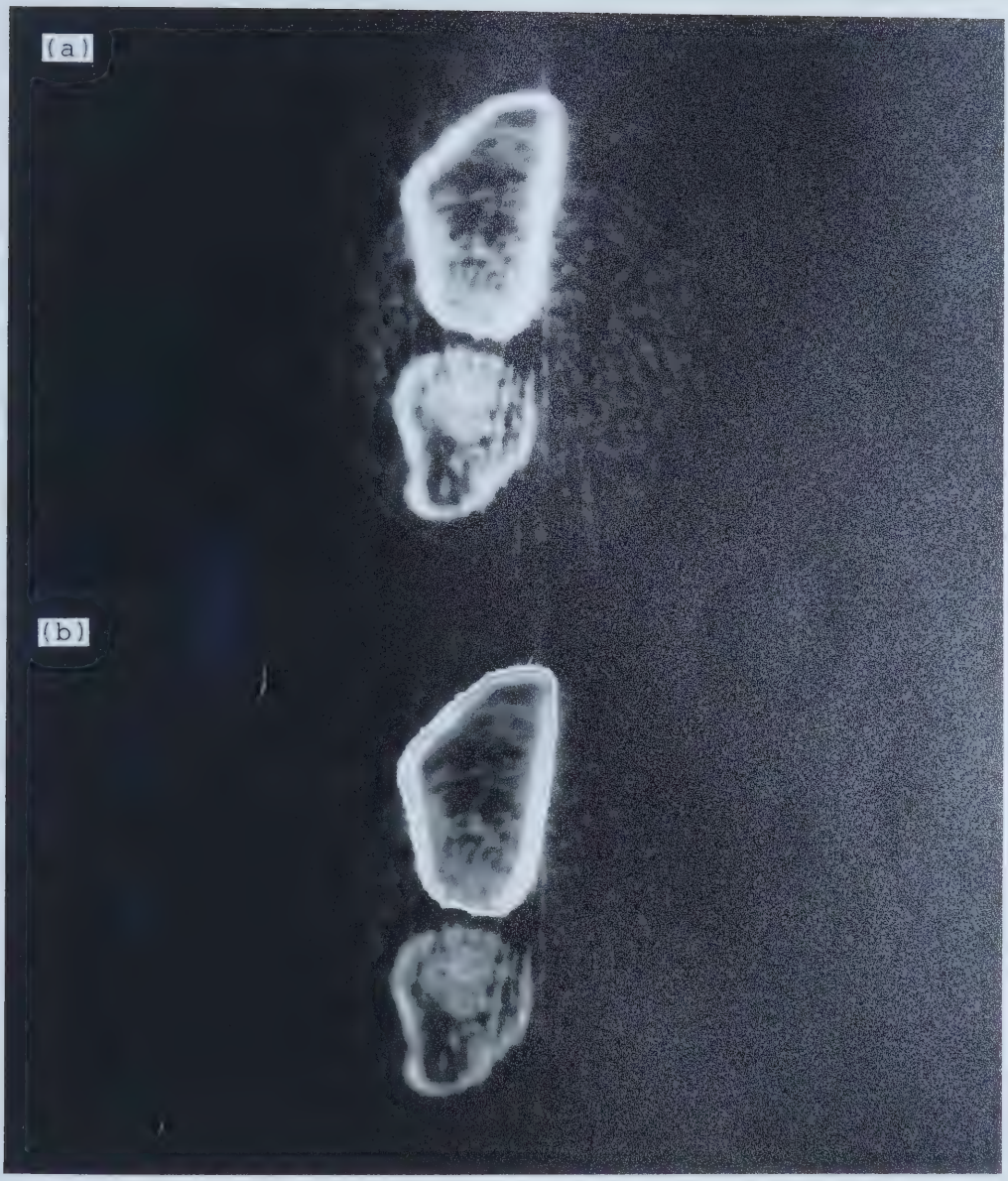


Figure 21. (a) An example of an input CT image for EDCORP; (b) The bone contour determined by EDCORP overlayed on (a).

half way between the ulna and the radius. To compensate for the ulna, the image is temporarily zeroed from the row of this center of intensity to the bottom of the image, assuming the orientation of the image is such that the ulna is roughly below the radius (see Figure 1). A new center of intensity of this zeroed image is determined, again using a threshold of 4000. Note that in a one-third distal scan, this new center of intensity will now be in the center of the radius; in a more distal scan or when the ulna has been 'erased', this new center of intensity will be slighty shifted away from the ulna but will now always be within the radius.

This center of intensity is taken as a first origin for construction of a set of radial profiles used to track the bone edge contour. The center of the determined contour is then used as a second origin for a new set of profiles to determine a better edge contour, and so on, until the origin location does not change significantly. At this stage, the origin should be at the true center of the radius bone and, therefore, the best origin for construction of the radial profiles. The resulting bone contour should then be the most precise.

In more detail, after the first guess for the origin is completed, REPEAT is set to True until the origin location stablizes or until a maximum of seven passes have been completed. For each pass, the first profile (PHI=360°) is constructed from the CT image. Each radial profile is a

nearest neighbor sampling of the CT image at 0.5 pixel increments along a radial vector at PHI. The entire first profile for each pass is cross-correlated with the kernel completely. The location of the maximum in the resulting correlation function is taken as the first edge point, and this point is marked in an EDGE image. Contour tracking begins with the first edge point and continues in a clockwise direction. Thus, the angle PHI for the next profile is decremented (1° was found to be sufficient).

For this and succeeding profiles in each pass, however, not the entire profile is cross-correlated with the kernel but only a portion within a window centered about the radial distance of the previous edge point from the origin. This limits the search for the next edge point and ensures good connectivity of the edge contour. The restriction to searching only within such a window is justified by the fact that the edge contour is quite smooth and does not change too rapidly in the radial direction. A window size of about ±5 pixels with respect to the radial distance of the previous edge point was found to work best. A smaller window did not provide enough latitude for the sharper corners of the cortical bone in some images. A larger window allowed for false edges to be tracked, especially into the trabecular bone when the end-plate was visible or into the ulna (the ulna's cortical-trabecular interface) when the ulna and radius were very close together.

To ensure a closed contour, each edge point is connected to the previous edge point by marking, in the EDGE array, a vector connecting the two points. The 1° spacing for the profiles usually ensures that the edge points are self-connected or adjacent, but short vector-interpolation is required in regions where the cortical bone displays sharper corners. After this connection procedure, PHI is again decremented for the next profile and the process of determining the next point in the edge contour continues until PHI reaches PHIEND=0°.

The last edge point found (at PHI=0°) should match the first edge point (at PHI=360°). If not, the contour is not closed, indicating a mis-tracking of edge points. In this case, the same edge point search technique is used but in the reverse direction, starting again at PHI=360° with the very first edge point found but now incrementing in a counter-clockwise direction. In this backtracking routine, a larger search window is used since the problem of non-connection may be due to corners of the cortical bone that are too sharp for the smaller window to handle (±15 pixels was found to be best). Often, however, the problem is due to the radius being too close to the ulna, in which case the incorrect edge contour has usually ended up following the ulna and then wandering out to the soft-tissue air interface. In this case, the larger search window used in backtracking is still small enough to find the correct cortical bone contour, but back only as far as the ulna.

Backtracking continues until re-connection with the old contour is achieved or until the starting angle PHI=720° is reached. In the latter case, EDCORP has failed to find a closed bone contour and, therefore, ends with an error message. Otherwise, if reconnection is made, the old portion of the contour that was by-passed by backtracking is erased in the EDGE array.

When EDCORP has successfully found the bone contour, the center of intensity of this contour is calculated considering all points marked in the EDGE array. If this center of intensity is effectively different (more than 1 pixel away) from the origin that was used in constructing the contour, then the entire tracking process is repeated in another pass using this center of intensity as a new origin for profile construction. If, however, the center of intensity is within 1 pixel of the origin for the current pass, then a further pass would only reproduce the same contour. Thus, the bone contour that has been found is EDCORP's best guess, and REPEAT is set to False to end the contour search.

The contour search is also ended if the origin does not stablize within seven passes, in which case the last contour is taken as EDCORP's best guess.

The final bone contour that EDCORP found from the image in Figure 21(a) is shown in Figure 21(b). This final contour was found in 2 passes.

At this stage, the final bone contour is filled and smoothed (using opening and closing) to create a radius overlay in exactly the same process as described in the algorithm for LCSKEL. The last computed center of intensity of the bone contour is used to center the original non-negative CT image in a 128x128 subimage. Again, as in LCSKEL, those pixels that correspond to pixels outside the radius overlay are negated. The resulting strip-negated subimage then effectively has the entire radius bone isolated and is compatible for viewing and further processing by Post-BOFI.

Upon output of this strip-negated subimage, the EDCORP algorithm ends.

4. Results of Comparing LCSKEL, EDCORP, and Other Contour Detection Algorithms

The algorithms of lower clipped skeletonization, edge detection by cross-correlation of radial profiles, and other contour detection methods (BOFI, Roberts, Rosenfeld) were compared using various sets of test images.

EDCORP, Roberts, and Rosenfeld(1-4) were compared on a set of ideal, blurred, and noisy test images based on a homogeneous square to determine accuracy and insensitivity to different types and amounts of noise. LCSKEL and BOFI were not included in this comparison. The results of this test showed that EDCORP was more precise and accurate than some of the best local edge detection algorithms in the literature.

Two sets (distal and proximal) of artificially degraded CT images of a patient's forearm were used to compare EDCORP, LCSKEL, and the BOFI system for precision and insensitivity to noise and artifacts. This comparison determined EDCORP to be superior, though LCKSEL was almost as precise. Both methods were significantly better than the BOFI system.

EDCORP and LCSKEL were further tested on a number of CT images of forearms taken over several months from a group of five subjects. The results from this patient set were compared with those results previously obtained using the BOFI system in a normal clinical routine. EDCORP was found to be superior in that it gave more reliable results with

much less user interaction. LCSKEL was not quite as reliable as EDCORP but still gave better results than BOFI.

4.1 Image Set Based on an Ideal Square

The objective of using this set of images based on an ideal square was to garner some quantitative measures to compare our edge detection algorithm with other well known methods. This comparison showed EDCORP was superior except when very high levels of noise existed. It was determined that the smoothing operation in EDCORP improved accuracy and sharpness of the detected edge contour; however, the structuring elements used in the opening and closing of the smoothing operation should be tuned to the object. It was also found that the kernel for cross-correlating the profiles need not be designed to match the expected edges, except in general character.

A study by T. Peli and D. Malah [32] used circles and squares, with added noise or blurring, to quantitatively compare several common local edge detection operators. These operators do not assume any *a priori* knowledge of the image; thus, they find to some degree all edges contained within the image. They are of a much more general class than BOFI, LCSKEL, or EDCORP which were developed for a specific image set (CT scans of the forearm). However, with only a few modifications, EDCORP was able to be applied to the same images that were used by Peli and Malah; unfortunately, LCSKEL and BOFI could not. Because it enhances ridges,

LCSKEL could not properly be applied to finding the edges of solid objects, and BOFI was not sufficiently robust to be used on these test images without major revisions to the algorithm.

Only a few results will be borrowed from Peli and Malah. In almost every case, the several edge detection operators that they tested worked best on the images based on the square rather than the circle. EDCORP, however, should perform better on circular objects since it makes use of radial profiles. Therefore, to bias results against EDCORP, only the image set based on the solid square was used in this comparison. In addition, Peli and Malah concluded from their study that:

> *For an image with a small amount of noise and not too slowly changing ramp edge, Robert's algorithm is the best one to use... If the image is noisy, Rosenfeld(1-4) is particularly suitable. At present, Rosenfeld's algorithm gives the best response, has the ability to detect curved edges, and is highly immune to noise.* [32]

Thus, only Roberts and Rosenfeld(1-4) from Peli and Malah's paper were included in this comparison with EDCORP.

The square image set was based on an ideal homogeneous square, 30x30 pixels, represented by a gray-level of 15 on a black background (gray-level 0). Included in the image set were a ramp edge square, the ideal square with added binary noise, and the square with added Gaussian noise.

The definition of the edge points for this comparison are the points (x,y) for any cross-section through the image as shown in Figure 22. The gray-level values and extent of the ramp edge are also shown. For noisy images, the location of the edge point is as shown, independent of how the edge appears.

The randomly distributed binary noise added to the square was at a 10% level. During the noise adding procedure, as each pixel in the image containing the ideal square was considered, there was a probability of 0.1 that the pixel was changed from a gray-level of 0 to one of 15, or from 15 to 0. There was a 0.9 probability that the pixel was left unaltered.

To create the Gaussian noise images, random values from a Gaussian distribution with zero mean and standard deviation of 3, 6, or 9 were added to the ideal square image. After addition of the noise, all values above 15 or below 0 were clipped; so these images did not display true Gaussian noise. However, consistency with the methods of Peli and Malah had to be maintained.

The quantitative measures Peli and Malah used to compare the performance of the edge detection algorithms were the following:

(R1) The weighted and normalized deviation of detected edge points from the ideal edge, as defined by Pratt [33]:

$$R1 = \frac{1}{I_N} \sum_{1}^{I_A} \frac{1}{1+\delta e^2}$$

where

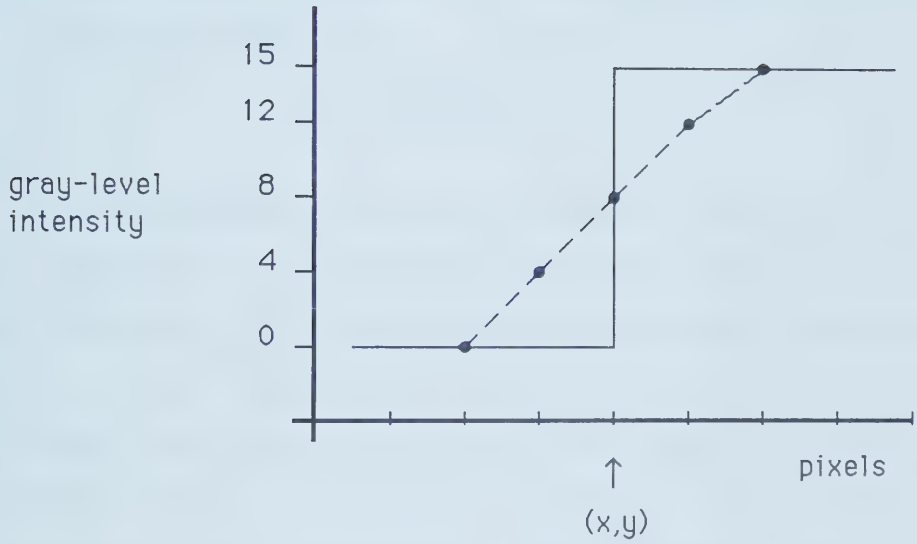


Figure 22. The definition of the edge points for ideal and ramp edges.

I_A = actual number of detected edge points

I_I = number of edge points on the ideal edge (116 for all these square test images)

I_N = maximum of I_A and I_I

δ = a scaling constant (taken as 1/9 for consistency with other work)

e = distance between a detected edge point and the ideal edge

(R2) Average squared deviation of detected edge points from the ideal edge:

$$R2 = \frac{1}{I_A} \sum_{1}^{I_A} e^2.$$

From the definition of R1, a perfect detection of the edge would give a score of R1=1.0 since e^2 would be zero and $I_N=I_A$. As detected edge points deviate further from the true edge, a lower value for R1 is obtained. Usually, more edge points are detected than actually exist; so $I_A>I_I$ and $I_N=I_A$. If instead, fewer edge points are detected than truly exist, suggesting a thin broken edge, then the value R1 will generally be lower for typical deviations. Unfortunately, because R1 is normalized, a detected edge may be quite thick (and even broken) but close to the ideal edge and yet have as high or higher an R1 value as another detected edge which is thin but not generally as close to the true edge. Also, as Peli and Malah confess:

> *R1 does not take into consideration any information on the distribution of the edge points along the edge.* [32]

R2 generally indicates the level of spatial noise in the resultant image. A score of 0 for R2 means no detected edge points deviate from the true edge (the detected edge may, however, be broken). A higher R2 value indicates either many pixels deviate slightly or a few detected edge points are very far from the true edge.

The algorithms discussed by Peli and Malah to which EDCORP was compared using the square image set were Robert's and Rosenfeld(1-4). Robert's algorithm [22] is simply the digital approximation to the gradient as discussed in Appendix I:

$$GRAD = \{(X(i,j)-X(i+1,j+1))^2 + (X(i+1,j)-X(i,j+1))^2\}^{1/2}.$$

As is commonly the case, Peli and Malah dispense with the square root and take the maximum of the two diagonal directional derivatives:

$$GRAD = \max\{|X(i,j)-X(i+1,j+1)|, |X(i+1,j)-X(i,j+1)|\}.$$

Rosenfeld's algorithm for edge detection [21,32,34] is introduced in Appendix I but will be described here in more detail. Basically, it is an extension to Robert's operation and involves computing the difference of average gray-levels in non-overlapping square neighborhoods on opposite sides of each pixel. Rosenfeld's operation can be implemented in various ways, but Peli and Malah used the following algorithm. For each point (i,j), compute the difference in average gray-level between neighborhoods of size $(2r+1)^2$ that meet at the point in the horizontal, vertical, and two diagonal directions:

$$E^{(r,H)}(i,j) = |f^{(r)}(i+r,j) - f^{(r)}(i-r-1,j)|$$

$$E^{(r,V)}(i,j) = |f^{(r)}(i,j+r) - f^{(r)}(i,j-r-1)|$$

$$E^{(r,DL)}(i,j) = |f^{(r)}(i+r,j+r) - f^{(r)}(i-r-1,j-r-1)|$$

$$E^{(r,DR)}(i,j) = |f^{(r)}(i+r+1,j-r-1) - f^{(r)}(i-r,j+r)|$$

where

$$f^{(r)}(i,j) = \frac{1}{(2r+1)^2} \sum_{m=i-r}^{i+r} \sum_{n=j-r}^{j+r} X(m,n).$$

Repeat these calculations for $r=r_1=1$ to $r=r_2=4$, thus using neighborhoods of size 3x3, 5x5, 7x7, and 9x9. For each point and each of the four fixed directions d (H,V,DL,DR) compute:

$$E^{(d)}(i,j) = \prod_{r=r_1}^{r_2} E^{(r,d)}(i,j).$$

Therefore, $E^{(d)}(i,j)$ will be large only if all of the $E^{(r,d)}(i,j)$'s are large. Involving large values of 'r' in the product reduces the effects of noise and ensures that only major edges are detected. Considering small 'r' values reduces the product $E^{(d)}$ at points not immediately near the actual edge, leading to sharp edge rendition.

Peli and Malah compared the above algorithm, Rosenfeld(1-4), to a variation, Rosenfeld(0-4) which used $r_1=0$ rather than 1. Results from Rosenfeld(1-4) were in almost all cases superior to Rosenfeld(0-4).

For this set of test images, EDCORP had to be slightly modified before it could successfully determine the edges of the squares. Since the square is only of gray-level 15, the threshold used in finding the center of intensity (the first guess for the origin of the profiles) had to be changed from 4000 to below 15. This worked fine for the images with the ideal and the ramp edge squares, but for the noisy images

the origin for the profiles had to be chosen manually. In these cases, the levels of noise were too high, and the squares covered too little an area of the images to significantly influence the center of intensity.

The results of the Robert's algorithm, Rosenfeld(1-4), and EDCORP when applied to the square image set are shown in Table 2. The ideal square with no added noise is listed as Gaussian noise image with a standard deviation of zero (σ=0).

Three sets of results are given for EDCORP. The first gives values for the detected edge before the contour is smoothed. The second set is after smoothing of the edge contour by an opening and a closing using, respectively, the large octagon and the square structuring elements as previously described in section 3.1.7. The final results are for EDCORP when the octagon structuring element is replaced by a square with sides of length 7 pixels. Recall that the octagon structuring element was designed for the opening to shave off slight protrusions from the bone edge overlay which is assumed to be smooth and rounded. For a square object, however, a roundish structuring element for the opening is not suitable, as it would round off the square's corners. A square structuring element is better tuned to the square object.

As seen from the table, EDCORP, before smoothing, did a perfect job in finding the edge of the ideal and ramp edge squares. This is remarkable considering that radial profiles

Table 2. Values of R1 and R2 for the various algorithms applied to the images in the square image set.

Algorithm \ Image	Gaussian Noise σ= 0	3	6	9	Binary Noise 10%	Ramp Edge
	R1 R2					
Roberts	.95 .52	.93 .54	.57 81.8	- -	.25 127.1	.90 1.11
Rosenfeld	.95 .52	.95 .52	.95 .55	.93 .80	.94 .70	.95 .55
EDCORP						
pre-smoothed	1.00 0	.95 .45	.90 1.18	.81 4.30	.91 1.12	1.00 0
post octagon smoothed	.92 .07	.95 .51	.93 .72	.88 1.74	.96 .48	.92 .07
post square smoothed	1.00 0	.99 .07	.96 .37	.90 1.67	.99 .15	1.00 0

through the square object were employed and that the kernel used was the same one designed for edge detection of the radius bone. This shows that the kernel need only resemble the edge of the object in character, not in detail. Since the pre-smoothed edge was perfect, and since the square structuring element is tuned to a square object, square smoothing did not affect these results. Of course, roundish smoothing cut off the square's corners reducing R1 and slightly increasing R2; but because the rest of the square contour was determined so well, these results are still comparable to those of the other algorithms.

Notice that for all noisy test images, the smoothing operation in EDCORP did not introduce significant error but, in most cases, actually improved the accuracy and reduced the noise of the pre-smoothed edge, even when the structuring element was not tuned to the image. Thus, smoothing of the edge contour found by cross-correlation of radial profiles is desirable for all cases examined.

Except when very high levels of noise were present in the image (σ=9, CV=9/15=60%), EDCORP, with smoothing, outperformed the better of general local edge operators. Both quantitatively and visually, the edges detected by EDCORP on all but the noisiest images in this test set were more accurate and much sharper.

4.2 Image Sets Based on Degraded CT Images of the Wrist

Two image sets were used to test the relative insensitivity of EDCORP, LCSKEL, and the BOFI system with respect variable noise conditions common to real images produced by gamma-ray CT scanners. This test showed that EDCORP was the more precise of the three methods, displaying the least sensitivity to noise.

The two image sets used were a distal set (close to the end-plate) and a proximal set (approximately 4 mm more proximal). Each set was based on a CT 'reference image' of a patient's wrist produced by our special purpose gamma-ray CT scanner for peripheral bone measurements [15] using a fresh ^{125}I photon source. The raw data for each CT reference image

is a collection of the photon count rates for the various transmission projections used to reconstruct that CT image [19]. To simulate the statistics associated with an older and weaker photon source, these raw data or photon count rates were reduced by a constant divisor, and appropriate levels of random noise were added. These modified data were then reconstructed, producing 'degraded' images of the same bone. In this way, successively noisier images of the exact same bone were produced. Due to repositioning error and variations in bone density, the exact same bone could not have been imaged periodically by re-scanning a patient's wrist several times.

The appropriate amount of noise to add to each point in the reduced projections was calculated considering photon counting statistics. Let one of the many transmission projections from which the reference CT image was reconstructed be denoted by $N(x)$, x being the spatial variable. A nine-point moving average filter [7] was applied to this projection profile to estimate the true photon count rate $I(x)$, without noise, at each point along the profile. Since photon count rates are described by Poisson statistics, the standard deviation $\sigma(x)$ of the expected error or the noise associated with this true count rate is $\sigma(x) = \{I(x)\}^{1/2}$ [35].

The observed photon count rate $N(x)$ can therefore be described as:

$$N(x) = I(x) \pm \sigma(x).$$

If the photon source was reduced in intensity by a factor d, then the count rate without noise would drop by the same factor and the observed count rate should become:

$$N'(x) = I'(x) \pm \sigma'(x),$$

where $I'(x)=(I(x)/d)$ and $\sigma'(x)=\sqrt{(I(x)/d)}=\sigma(x)/\sqrt{d}$ according to Poisson statistics. The term $\sigma'(x)$ represents the standard deviation of the noise associated with this reduced count rate. The actual number of counts to be added must be governed by this standard deviation and the associated Poisson statistics. Therefore, a random noise sample (q(x)) is added to each of the reduced projection values ($N/d = I/d \pm \sigma$) in order to bring the level of noise to the correct amount

$$\sigma'(x) = \sigma/\sqrt{d} = \{\frac{\sigma^2}{d^2} + q^2\}^{1/2}$$

where according to error propagation, the error in the sum of two values is the square root of the sums of the squared errors of each value [36]. Solving for q gives:

$$q(x) = \frac{\sigma(x)}{d} (d-1)^{1/2}.$$

Thus, a random sample from a Gaussian distribution with standard deviation q(x) and zero mean was added to each point in the reduced projection N(x)/d, giving an approximation to the original projection as if it were measured using a weaker photon source.

To calculate the amount of noise to be added, using a Gaussian distribution is a good approximation to the Poisson distribution for larger count rates (N(x)/d) [35]. For the few lower count rate values present in a complete set of

projections (≃200 out of 79296 count rate values), the difference in the two distributions is insignificant. After the randomly chosen values, which are usually small, are added to the reduced projection values, the sum is rounded to the nearest integer.

It should be noted that whenever a random sample from the Gaussian distribution would have resulted in a negative value when added to the reduced count rate, the noise sample was not added, leaving the divided count rate unchanged. This modification is seldom required (≃100 out of 79296 values) but actually leads to a better approximation of a Poisson distribution.

An alternative procedure of simulating an older photon source would have been to choose as the reduced projection value a random sample from a Poisson distribution with mean equal to the output of the nine-point average filter divided by the divisor, I(x)/d. This would have produced a projection with better (Poisson) statistics. But using the average value in an estimate for the reduced photon count rate, instead of just as a means to estimating $\sigma(x)$, would have led to blurring of the degraded CT image. This blurring is an effect not naturally produced by a weaker photon source.

The described method of adding noise to CT images to simulate those produced using a weaker photon source was applied to a distal and a proximal reference image of the wrist of a 62 year old osteoporotic woman. For each

reference image, a set of 10 noisier images were created with count rate divisors (d) ranging from 2.0 to 6.5 in steps of 0.5. The noisiest image corresponds to an ^{125}I source of about 2.7 half-lives old. The distal and proximal reference images and an example from each set of the corresponding degraded images (d=6.5) are displayed in Figure 23(a).

In addition, artifically degraded images with count rate divisor d=6.7 were compared with similar CT images of the same wrist region produced about 5 months later using the same, but now older, photon source just before it was replaced. This old ^{125}I source gave open count rates that were down by a factor of about 6.7 compared with the fresh photon source. Visually, the artificial and the measured degraded images showed similar levels of noise and artifacts. Table 3 shows that in corresponding regions of soft-tissue (area=1313 pixels), the standard deviations about the mean pixel value were similar in the artificial and actual images.

Each contour detection algorithm was applied to all the images in both image sets. The radius bone overlays resulting from each of the degraded images in a set were compared with the overlay resulting from that set's reference image. For each noisy image, the positive and negative pixel differences were considered. The positive pixel difference is the number of pixels in the radius overlay determined from the noisy image that are not also in

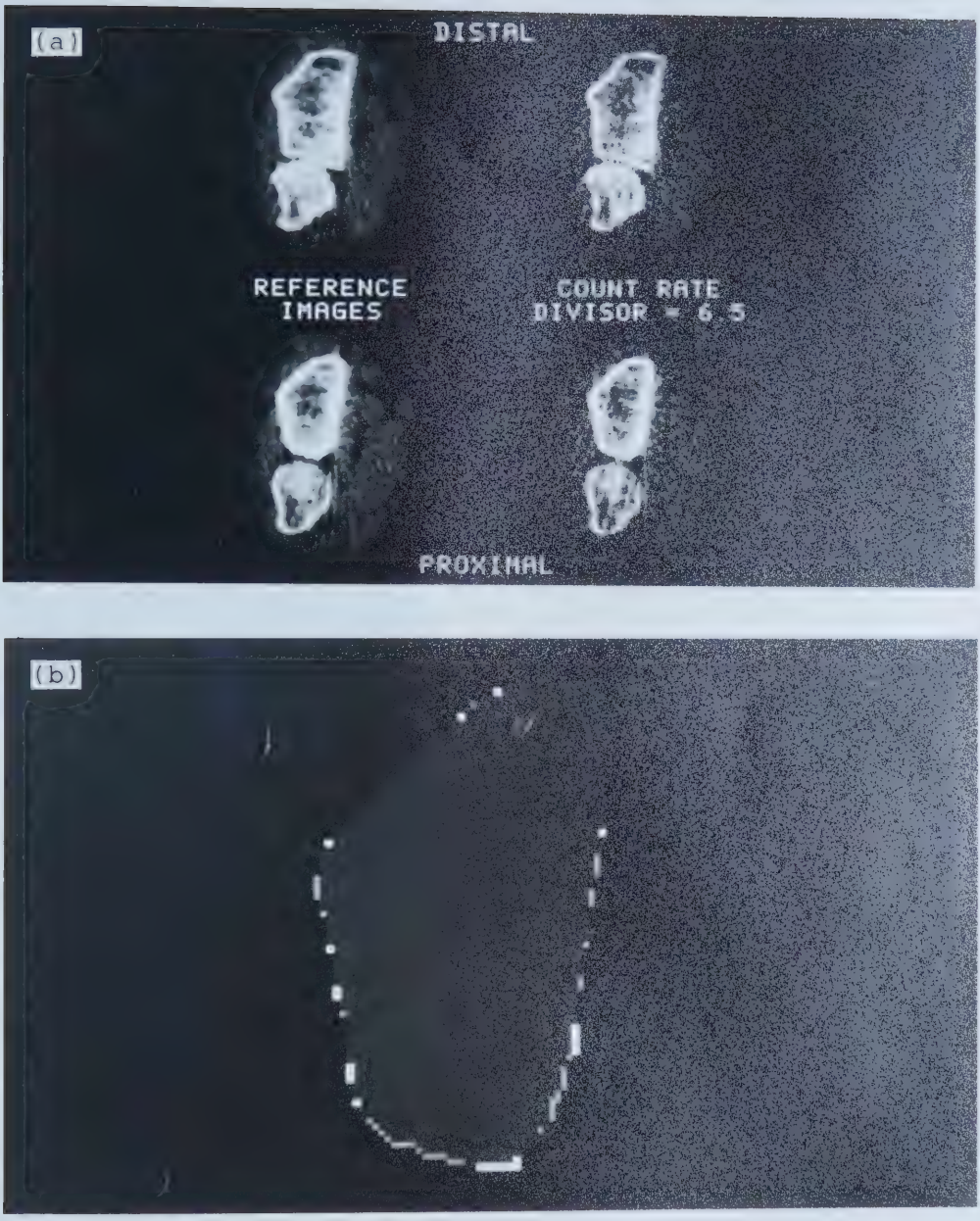


Figure 23. (a) Distal and proximal CT images of the wrist: (left) original images; (right) maximum noise added. (b) Radius overlay (dark gray + white) determined by EDCORP from a degraded (d=6.0) CT image superimposed on the bone overlay (dark gray + gray) from the corresponding reference image: white - positive pixel differences; gray - negative pixel differences; dark gray - pixels common to both overlays.

Table 3. Comparison of mean CT number and standard deviation (σ) in a soft-tissue region (area=1313 pixels) of actually and artificially 'degraded' distal and proximal CT images of the wrist. The artificial images were produced from CT 'reference images' with a count rate divisor of d=6.7. The actual noisy images were imaged with an old photon source producing count rates down by a factor of approximately 6.7.

Distal Images	Mean CT Number	σ
reference image (no noise added)	2780	255
artificially degraded image (d=6.7)	2663	514
actual noisy image	2718	584
Proximal Images	**Mean CT Number**	**σ**
reference image (no noise added)	2773	250
artificially degraded image (d=6.7)	2683	536
actual noisy image	2752	592

the overlay from the corresponding reference image; the negative pixel difference is the number of pixels not in the overlay from the degraded image that are in the overlay from the reference image. Figure 23(b) illustrates an example of the radius bone overlay (dark gray + white) of a degraded (d=6.0) proximal image superimposed on the overlay (dark gray + gray) determined from the corresponding reference image. The number of white pixels is the positive pixel difference whereas the number of gray pixels is the negative difference. For each noisy image in both the proximal and the distal image sets, the positive and negative pixel differences were plotted (Figures 24 and 25).

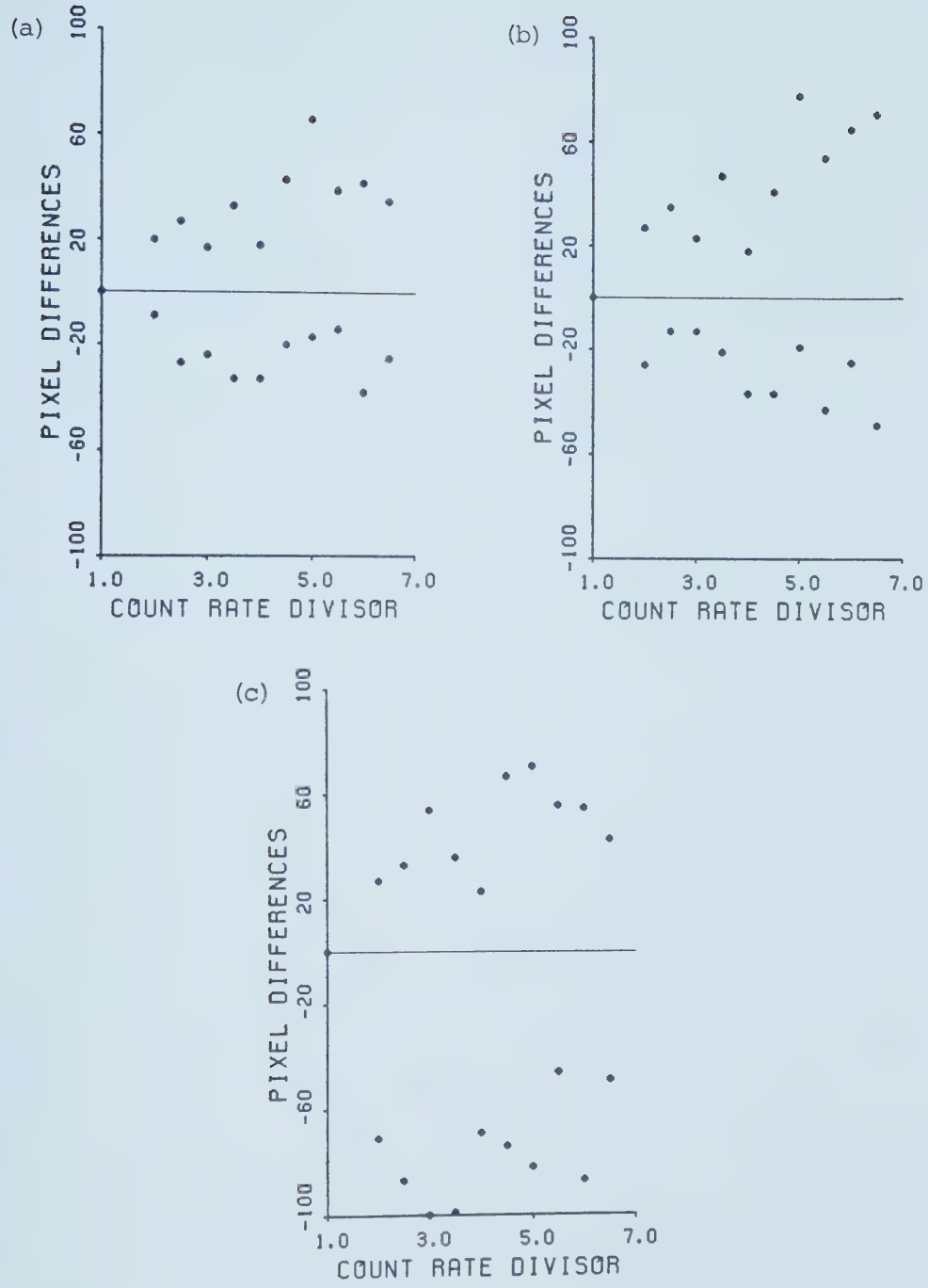


Figure 24. Positive and negative pixels differences for the radius overlays of the 10 noisy images in the distal image set produced by: (a) EDCORP; (b) LCSKEL; (c) BOFI system.

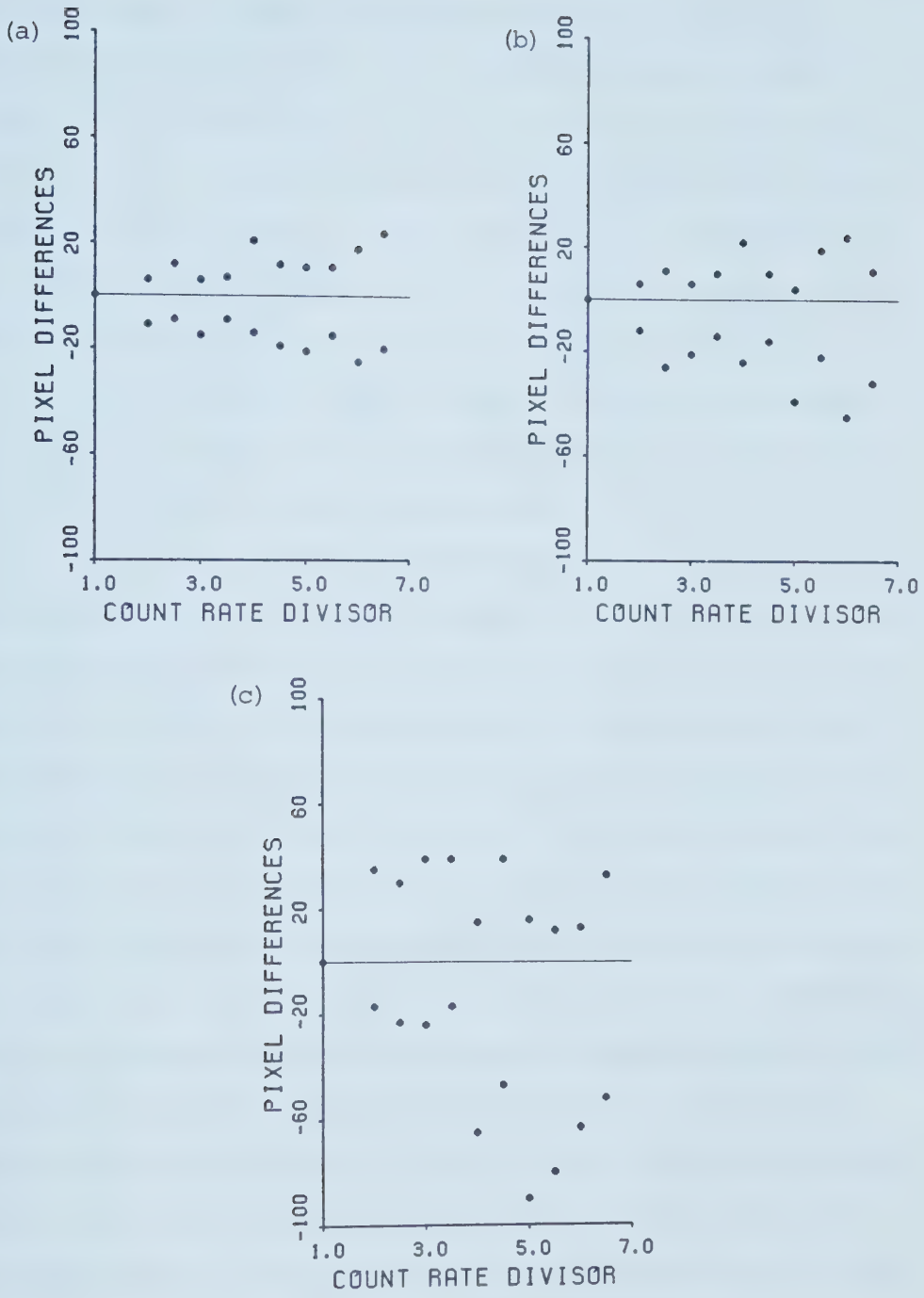


Figure 25. Positive and negative pixels differences for the radius overlays of the 10 noisy images in the proximal image set produced by: (a) EDCORP; (b) LCSKEL; (c) BOFI system.

Note that the positive pixel difference less the negative pixel difference gives the difference in area between the two overlays. A plot of the pixel differences, however, gives a better indication of deviation than simply noting the differences in total area between overlays. Two overlays may have the same area but differ in shape, giving equally large positive and negative pixel differences.

For completeness, Table 4 gives the mean bone areas calculated from each set of overlays and the corresponding standard deviations about these means.

As the plots and Table 4 indicate, the interactive BOFI method exhibited the expected large random pixel differences as the BOFI operator visually under- or over-corrected the bone contour using the trackball and cursor system. Even though the cortical bone is generally better defined in the proximal images, the manual contour correction in BOFI led to a larger variance in bone area for this set rather than the distal image set. The LCSKEL algorithm was significantly more consistent in isolating the radius; however, EDCORP was superior, especially for the proximal image set. Both LCSKEL and EDCORP tended to produce only slightly larger pixel differences as the levels of noise in the images increased, and both were more consistent for the proximal image set where the cortical bone is more dense and generally thicker. With respect to insensitivity to the variable noise conditions common to the type of CT images considered here, EDCORP was the most consistent method of contour detection.

Table 4. Mean and standard deviation (σ) for bone overlay areas. The data were obtained from a distal and a proximal set of 10 'degraded' images of the radius subjected to each of the 3 contour detecting algorithms.

Algorithm	Bone Image Set	Mean Mask Area [pixels]	σ [pixels]
EDCORP	Distal Proximal	4848 3486	18 6
LCSKEL	Distal Proximal	4148 2894	22 11
BOFI System	Distal Proximal	4895 3434	26 37

4.3 Image Set Representing a Clinical Study

The purpose of this set of images taken from a bone study of five patients was to compare EDCORP and LCSKEL with the currently employed contour detection system, BOFI, in a clinical setting.

The image set consisted of 676 useable CT images accumulated over about seven months from five different patients. At each wrist scanning session about six CT images at 2mm increments were usually obtained. All images judged unacceptable due to improper placement of the forearm, motion artefacts, or hardware failures were ignored. The remaining 676 images provided a typical cross-section of the types and quality of scans expected in a peripheral bone density study of post-menopausal, osteoporotic women.

The images in this set where submitted to the BOFI procedure. In Pre-BOFI, a subimage for each image had to be

selected by using the trackball to center a frame over the radius. About 50% of these subimages included a portion of the ulna. In these cases, the ulna had to be 'erased' using the trackball to trace a line separating the ulna from the radius.

The subimages were then acceptable for input into Auto-BOFI which median and Laplacian filtered each subimage, found the appropriate zero-crossings of the Laplacian image, and marked the resulting radius contour via a strip-negated subimage. These subimages were then submitted to Post-BOFI for visual inspection and any necessary correction of the determined bone contours. Of the 676 radius contours determined by Auto-BOFI, 96% required manual correction using the trackball system. This visual correction is very subjective and varies with different operators and various display intensity levels.

The same 676 images were submitted to EDCORP. In an average of 56 CPU seconds (on a VAX 11/750 computer system), EDCORP automatically determined a bone contour, framed a subimage, and produced the strip-negated version for input to Post-BOFI. Visual inspection in Post-BOFI passed 83% of this initial run-through as not requiring contour correction.

The remaining 17% of the CT images were subjected to modified Pre-BOFI stage to manually 'erase' the ulna using the trackball and cursor system. In this modified Pre-BOFI, no manual subimage framing was required; only a separation

line in the original CT image had to be drawn to remove the ulna. Then, rather than extending the separation line pixel values column-wise to the bottom of the image, a single replacement value of 3999 was set for all pixels below the separation line. This value of 3999 is less than the 4000 threshold used to determine the profile origin; so these pixels cannot contribute to the center of intensity calculation. The value is usually high enough so that a false dropoff is not created at the separation line. Such a dropoff would create a false edge which, when sampled by a profile, would correlate highly with the kernel and confuse the EDCORP algorithm.

The 117 images with the ulna suppressed were then resubmitted to EDCORP, and the resulting subimages were passed to Post-BOFI for inspection. Only 38 of these images were deemed to have an incorrect bone contour on this second submission. Thus, only about 6% of the 676 CT images submitted to EDCORP ultimately required manual correction of the radius contour.

Of these 38 images for which EDCORP twice failed to determine the correct bone contour, about half were due to inclusion of the high density growth-plate within the image, which caused the tracking routine to fail. If greater care was taken to properly position the most distal scans just short of the growth-plate, EDCORP would have ultimately had a 97% success rate in contour detection.

LCSKEL was determined inferior to EDCORP for this application for a number of reasons; therefore, the testing of LCSKEL was not carried out to the same degree as for BOFI and EDCORP. Because of the large number of thinnings required in both lower sequential thinning and end-point clipping, LCSKEL takes approximately 35 minutes run-time on our VAX 11/750 system. (EDCORP requires less than a minute). Compared to EDCORP, LCSKEL was not as insensitive to noise. Of particular importance, since LCSKEL does not find the actual edge of the bone but rather the cortical ridge contour (usually running through the center of the cortical shell), the accuracy of LCSKEL may decrease during serial studies. Osteoporosis does not usually affect the outer edge of the cortical bone, but it may reduce the volumetric density of the inner cortical bone adjacent to the trabecular bone. This would lead to a physical thinning of the cortex over time and, therefore, a systematic outward shift of the cortical ridge contour detected by LCSKEL. Also, LCSKEL requires that the ulna somehow be removed from the image, either before or after ridge detection, since it maintains all closed ridges in the image. It was therefore determined that LCSKEL, by itself, could not compete with EDCORP for this application.

Only a 60 image sample of the 676 test images were submitted to LCSKEL to estimate its relative performance, 12 from each of the five subjects. All of these images first had their ulna manually 'erased' in a modified Pre-BOFI

procedure. Here the full CT image was kept and no manual subimage framing was required. Then in an average of 35 CPU minutes, LCSKEL automatically determined the radius contour and produced the framed strip-negated subimage which would be suitable for input into Post-BOFI. Visual inspection of the resulting cortical ridge contours determined that about 72% were acceptable; 28% would require manual correction. Based upon these preliminary results and the results obtained with EDCORP, it was determined that LCSKEL did not warrant further investigation.

The results of applying BOFI, LCSKEL and EDCORP to this clinical image set are summarized in Table 5. EDCORP clearly requires less user interaction and is more reliable than either the BOFI system or LCSKEL. Since manual correction is seldom needed, the bone contour produced by EDCORP is likely more precise.

LCSKEL did exhibit slightly better results than the BOFI system in this test case. Even though both procedures require removal of the ulna, it is only a one step process in LCSKEL. BOFI requires framing in all and ulna erasing in 33% of all cases. Based on the images examined, fewer of the bone contours determined by LCSKEL would require manual correction, generally resulting in better reproducibility and less operator time than processing by BOFI.

In general, the radius contours found by EDCORP compared well with those ultimately determined by the BOFI system. An example of some of the plots of bone area versus

Table 5. Comparison of the performance of EDCORP, LCSKEL, and the BOFI system applied to a clinical set of 676 CT images of the wrist from five subjects. (Results for LCSKEL are based upon a 60 image subset.)

	BOFI	LCSKEL	EDCORP
images which required manual subimage framing	100%	0%	0%
images which required ulna removal by erasing	33%	100%	0% initially 17% for resubmission
images which required manual contour correction	96%	28%	6%

relative axial scan location calculated from the BOFI system and EDCORP output images for different measurement sessions are shown in Figure 26. Contour corrections via Post-BOFI were made for the BOFI results when necessary. These plots confirm that any differences between the bone overlays produced by the two algorithms are very small.

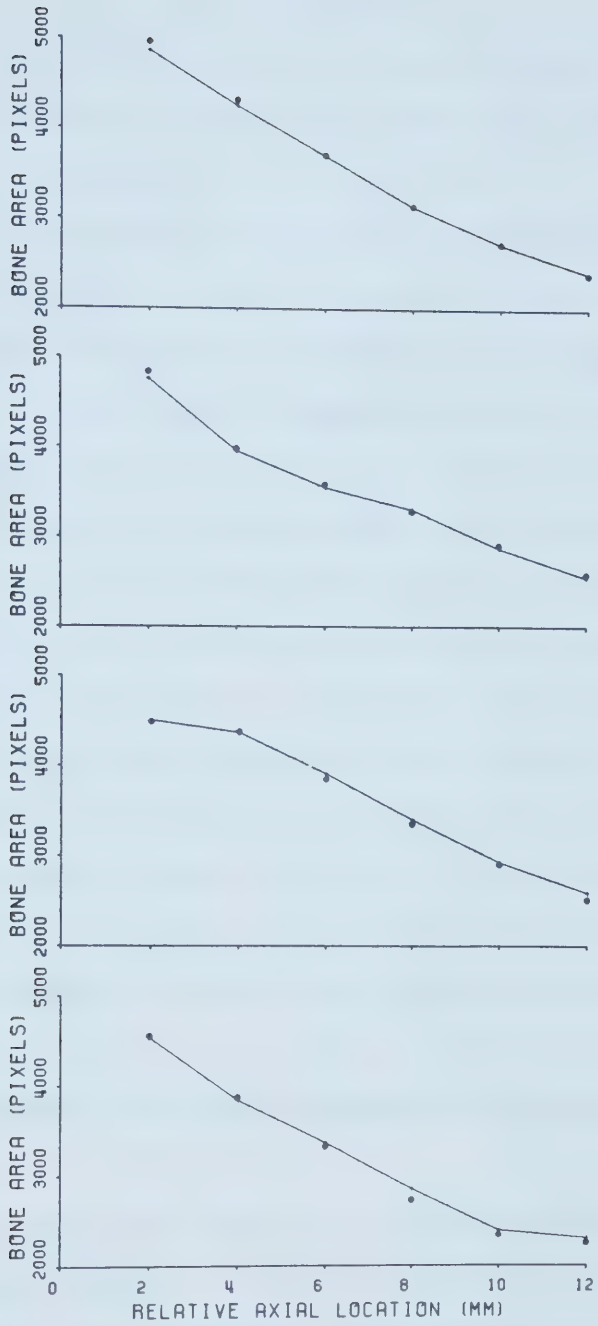


Figure 26. Plots of radius area versus relative axial location calculated from contours determined by BOFI and EDCORP after manual correction of the BOFI contours when necessary. The EDCORP data points (+) for each measurement session are connected by straight lines. Solid dots are BOFI data points.

5. Discussion

Both EDCORP and LCSKEL compared well with other contour detection algorithms, although they each have their own strengths and weaknesses.

By design, the major strength of EDCORP is its insensitivity to low or moderate levels of noise. This was confirmed by the performance of EDCORP on the square image set (Table 2) and the real CT images of the wrist degraded by the additions of noise (Figures 24 and 25). A reason for this is the bandpass-filtering effect of the kernel used in cross-correlation. Generally, noise spikes across the ramp edge in a profile are of a very high spatial frequency and are filtered out by the kernel. The ramp edge itself is a low frequency phenomenon, and the kernel tends to sharpen such an edge by suppressing some of the lower frequencies. Altering the kernel in small details, but maintaining its general character, does not greatly affect its bandpass nature. This suggests why the same kernel performs well for the various profiles within an image, for the many different CT images, and even for other classes of images such as the squares.

However, when high levels of noise are present, the noise spikes may combine to create relatively broad false edges which may have frequencies predominately within the kernel's pass band. This was the case of the square image with Gaussian noise σ=9 (see Table 2). The effect was compounded by the clipping of all values over 15 and under 0

in the creation of this noisy image. These false edges may correlate well with the kernel, leading to mistracking of the contour. Fortunately, such high levels of noise are not common in our CT images. The noisiest images used for calculating the plots of Figures 24 and 25 contained levels of noise corresponding to a ^{125}I source 2.7 half-lives old. Generally, the source is changed every 1.5 to 2.0 half-lives (4 months), corresponding to a count rate divisor of 4.0 or less.

Other attractive features of EDCORP are its automation and reliability. Due to constrained contour tracking, tests for connectivity, and the backtracking process, EDCORP usually resulted either in a good contour selection (83% of the time) or else in obvious failure. Not much time was required to visually inspect the isolation of the radius to decide if it was acceptable. The proximity of the ulna to the radius had to be dealt with in only 17% of the images, and correction of the contour was required in only 6% of the images on average. The reduction in operator involvement helps to streamline the processing of the large amount of data required by clinical bone studies.

A common reason for the failure of EDCORP in determining a good bone contour is its improper selection of the very first edge point (along the PHI=360° profile). Usually the cause is inclusion of the high density growth-plate within the image (interior to the cortical shell) whose edge correlates well with the kernel.

Two other causes of mis-tracking are the presence of sharp corners along the cortical contour and the proximity of the ulna to the radius. Sharp corners are not well sampled by the radial profiles, and the change in radial location of the bone edge around such a corner may be too great for the search window size. The problem corner area is usually the upper right portion of the radial cortex for right hand distal scans (see Figure 1).

To minimize mis-tracking, the edge point search begins at PHI=360° where the cortex is usually most dense, giving good selection of a first edge point. Tracking continues in a clockwise direction. When a sharp corner at the top right causes improper tracking, a short distance of backtracking (with a larger search window) easily reconnects with the old contour just at the corner. This is graphically illustrated in Figure 27(a). If a counter-clockwise direction were used, the top right corner would often lead to mis-tracking for the many remaining profiles. Backtracking in the reverse direction would then have to proceed between the ulna and radius, but the larger search window necessary to round the problem corner may be too great to successfully ignore the ulna (see Figure 27(b)). The same problem occurs if tracking starts at 180° (where the cortex is almost as dense as at PHI=360°) and proceeds clockwise to -180° as shown in Figure 27(c). If the contour search begins at -180° and continues counter-clockwise, to 180°, then the bottom portion of the contour near the ulna is usually tracked well; however, the

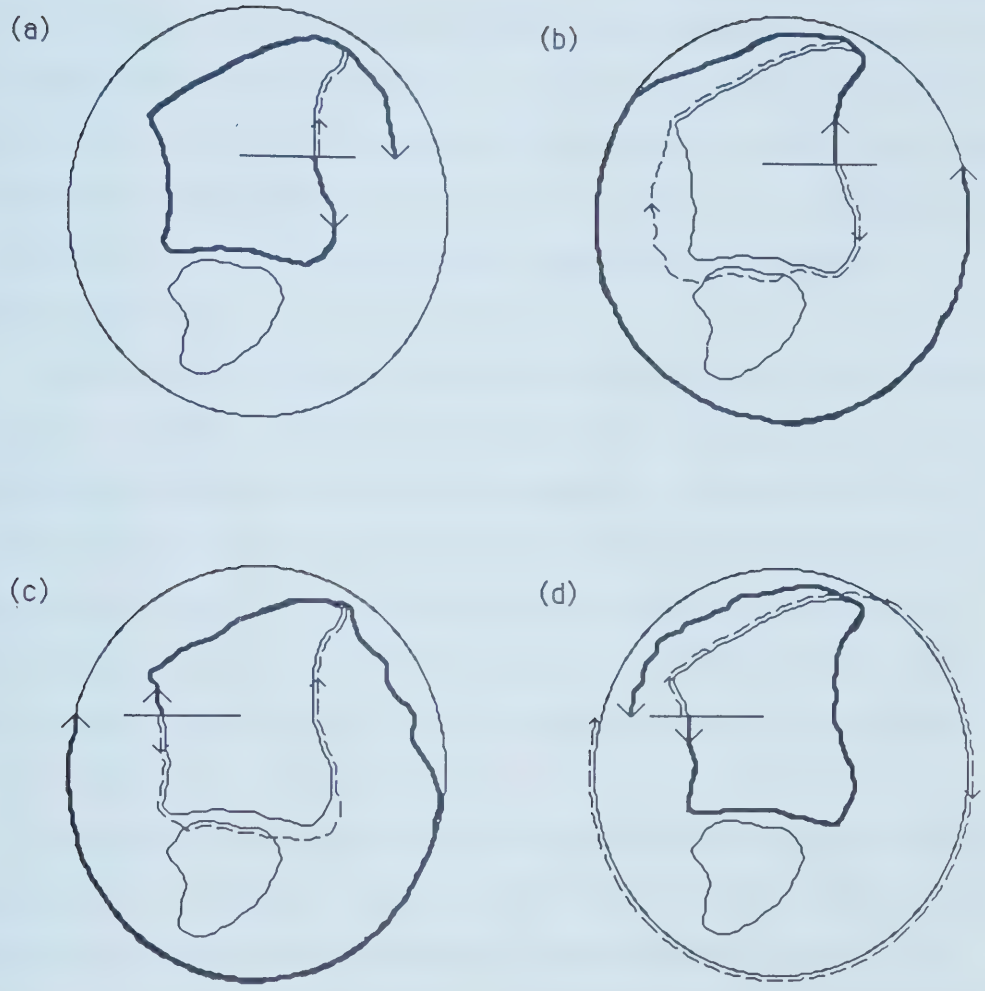


Figure 27. Different contour tracking directions and starting locations. Example (a) minimizes mis-tracking for right-hand scans.

top right corner may again cause mis-tracking. Upon backtracking, the search window is now too wide to exclude tracking of the nearby soft-tissue-air interface at the top of the CT scan (see Figure 27(d)). The current tracking scheme minimizes the chance of error, but mis-tracking still occurs (6% on average). Dynamic programing, which allows for tracking of many rather than just one contour, is discussed in Section 7.1.3 along with other possible solutions to these problems.

Fortunately, from a developmental point of view, EDCORP is very flexible. Changes or refinements are easily implemented. The algorithm is adaptable for application to image analysis problems other than those of bone contour detection. Kernals can be designed for edges of different character, profile selection may be other than radial, and the contour tracking criteria may be altered, all to suit other classes of images.

Because of the rather strict framework of Mathematical Morphology, LCSKEL is not as flexible as EDCORP. Neither is LCSKEL as insensitive to noise or artifacts, though it is more reliable than Auto-BOFI alone and, due mainly to its automation, more precise than BOFI in conjuction with manual contour correction (see Figures 24 and 25).

Since thinning is based on principles of connectivity, any breaks in a ridge within the image will lead to a poor skeleton. This is a major weakness of LCSKEL for application to our CT images of the forearm. In the more distal scans,

the cortical shell is often thin and of low density. Sometimes, small breaks in the continuity of the cortex, exacerbated by noise, are of a lower density than nearby ridges wandering through the trabecular bone (especially on the distal scans which are quite close to the growth-plate). Thus, the ridge determined by LCSKEL may exclude a large portion of the cortical contour. In comparison, EDCORP, by using profiles to sample the cortex, is unlikely to encounter these breaks directly. EDCORP is also forced not to accept edges outside the search window; therefore, such breaks are successfully spanned.

Another weakness of lower clipped thinning is the enhancement of all closed ridges. Further pixel classification, such as by thresholding, is required to extract the appropriate ridge from the background of generally lower ridges. Sometimes, due to dips along the cortical ridge, the ridge is not exactly preserved in thinning; instead, portions of it are reduced in value. This makes it difficult to use a simple threshold to segment the entire cortical ridge from the background.

This problem of extracting enhanced contours is common to many edge and contour detection techniques (see Appendix 1). However, LCSKEL does have an advantage in that it reduces the ridge contours to a single-pixel width. Many edge detection methods give a high response value to all pixels within a broad strip about each edge in an image; consequently, classifying the correct contour in such cases

is even more difficult.

Other methods of digitally thinning images do exist. An example by Rosenfeld [8] gives quite different results than lower sequential thinning by {Li}. Consider thinning of binary images or sets. Rosenfeld defines 'simple points' and four types of 'border points'. A north border point, for example, is a point in the set with a 0 neighbor to the top or north. A point P is simple if those of its 8-neighbors which also belong to the set (1's) form exactly one connected component without necessarily containing P. His thinning algorithm is the following: border points of the set are removed (zeroed) if and only if they are simple and not end-points; this is done successively from the ...north, south, east, west... until no further change occurs.

Notice, in contrast, that the five members of the sequence {Li} that can remove points which are north border points do so because such points are simple; however, not all simple points are removed by structuring elements included in {Li}. For example, in a portion of the image:

```
0  0  0
0  1  0
0  1  1
```

the central (north border) point is simple and would be removed by Rosenfeld's algorithm, but it is not removed by thinning with {Li}.

By Rosenfeld's definition, north border points may actually be on the west or east side of the set, and similarly for other border directions. Thus, to ensure that

the resulting skeleton is centralized in the set, Rosenfeld must employ an alternating north, south, east, west sequence, rather than the rotational sequence incorporated into {Li}. An alternating sequence for {Li} would not generally produce significantly different skeletons than the standard spiral thinning produced by the rotational sequence.

Because Rosenfeld does not use structuring elements, a seemingly advantageous property is that his thinning can take place from within small holes surrounded by the set X so that the resulting skeleton is roughly midway between these holes and the outside of the set. In our thinning, holes with a linear dimension of less than 3 pixels, or the size of the 'off' portion of the structuring element, are effectively ignored since the o's of the structuring element cannot fit inside these holes. Thinning then takes place only from the outside of the set inward to these holes. A simple example of this effect is seen in Figure 28 where the results of the two different thinning procedures are shown. These properties can be carried over to gray-tone images. Thus, for images with noise (such as our CT images of the wrist), small holes due to graininess of the image are ignored by thinning with {Li} to the extent that the skeleton just skirts them. Rosenfeld's method of thinning would result in a skeleton with much larger loops, emphasizing the graininess of the image.

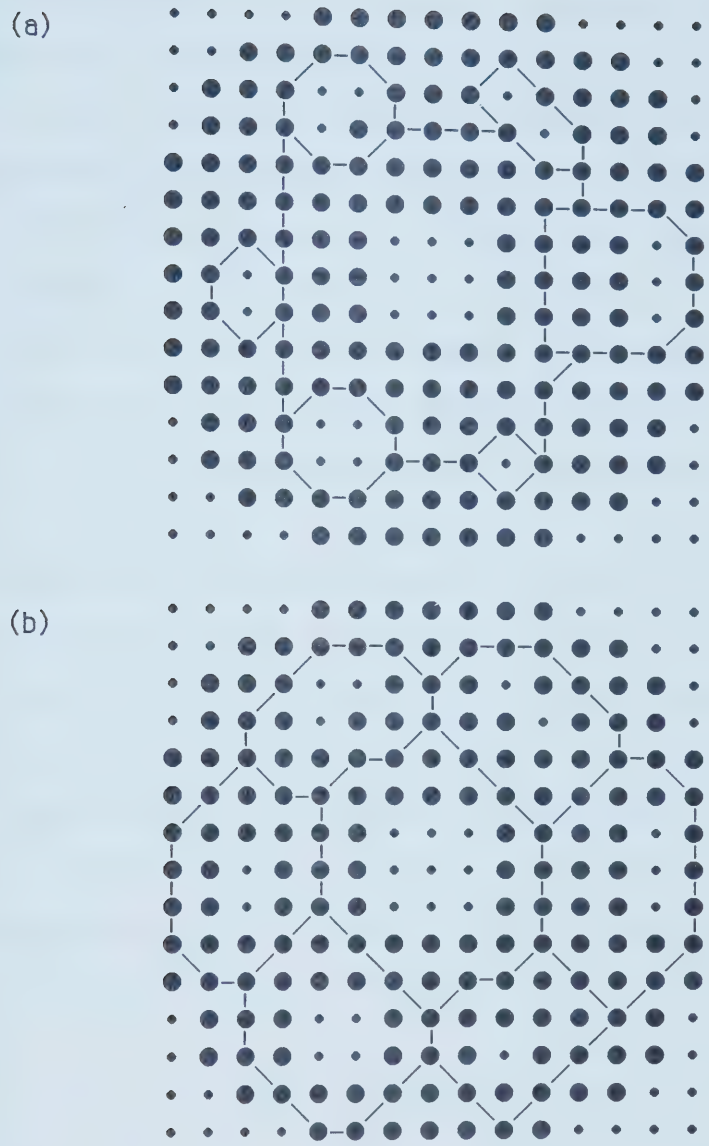
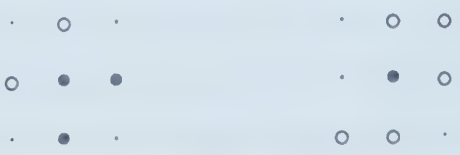


Figure 28. A comparison of Rosenfeld's thinning and sequential thinning by {Li} for an image set with small holes: (a) Sequential thinning by {Li} and (b) thinning using Rosenfeld's algorithm, result in the set of pixels connected by the lines.

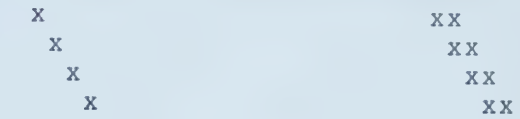
Finally, some discussion of the grid used in Mathematical Morphology is in order. Throughout his text, Serra uses the "morphologically richer" [26] hexagonal grid to digitize his images and define his structuring elements. The advantages of a hexagonal grid over an 8-connected grid include rotational symmetry in 6 rather than 4+4 directions, the requirement for only one rather than often two different forms of the same structuring element, and the ability to better approximate disks. These advantages are especially important in thinning procedures where connectivity is involved.

Notice that by rotating a square grid by 45°, it becomes a hexagonal grid, albeit with unequal spacing between a central point and its six neighbors. However, since it is the structure of the images with which Mathematical Morphology is generally concerned, it was decided to try structuring elements based on this pseudo-hexagonal grid, for example the thinning structuring elements L1 and E1 respectively:



These and their corresponding sequences (of six rotations) were used in LCSKEL to produce the lower clipped skeleton of a few CT images of the wrist (the images themselves are not actually altered in any way, only viewed from a 45° rotation). Surprisingly, no significant differences in the

final ridge contour were apparent compared with using the standard 8-connected structuring elements; the contours matched everywhere to within ±1 pixel. However, the pseudo-hex processing did result in a preponderance of pixels on the major-axis sides (northeast and southwest) of the bone:



8-connected pseudo-hex

since there was no structuring element configuration to remove such points. Such contours are, however, still single-pixel width as defined on the pseudo-hex grid.

One might suspect that the pseudo-hex thinning process would run 8/6 times faster than the 8-connected thinning due to the fewer rotations involved in its lower sequential thinning sequence; however, it was found that they were actually slower. This is due to the larger number of cycles of the pseudo-hex sequence required for thinning or clipping all the points in an image; the 8-connected sequence removes (lowers) more points per cycle. (A cycle is one set of 6 or 8 rotations.) For example, a single vertical line of 'set' pixels has three pixels removed from the bottom per cycle of 8-connected clipping with {Ei} and only two per cycle for pseudo-hex clipping. On this basis, it may also be argued that the pseudo-hex sequential thinnings produce contours which are differently centered within the cortical shell. Any differences, however, were negligible.

The problem with this pseudo-hex grid becomes very apparent in erosions and dilations. For example, in the 8-connected grid, a 3x3 square structuring element (which is an approximation to a unit disk) can be used to dilate an object equally on all sides. In the pseudo-hex grid, however, the unit disk is represented by an irregular hexagon:

Dilation of a similar object by this structuring element would not dilate the object into the northwest/southeast diagonals as far as other directions.

6. Conclusions

Two modular, automated contour detection algorithms were developed for isolating the radius in CT images of the forearm. Precise determination of the bone contour is required in serial studies of bone density using CT to ensure that the identical volume of bone is evaluated on each occasion. Automation in this process is required to increase reproducibility and decrease the time required for analysis of the large amounts of data common to most clinical investigations with this method.

The algorithm of lower clipped skeletonization provides a precise contour of the radial cortical shell when the imaged cortical bone is dense, well defined, and without breaks. For such images, LCSKEL is relatively insensitive to low or moderate levels of noise. Unfortunately, in a clinical measurement situation, many CT images of the forearm exhibit low density cortical shells with poor connectivity. When such cases are compounded by image noise, LCSKEL usually cannot determine the correct bone contour. In all cases, however, the ulna must be removed either before or after processing by LCSKEL; if done manually, much operator involvement is required.

Edge detection by cross-correlation of radial profiles is more precise and much more reliable in its determination of the bone contour than LCSKEL. EDCORP is also more precise and less sensitive to low or moderate levels of noise than either BOFI or some of the best common local edge detection

algorithms in the literature. The cross-correlation kernel need only reflect the character of the edges to which it is matched, and smoothing by opening and closing the bone overlay improves results if the structuring elements are properly chosen. In the majority of cases, a closed, smooth, discrete contour of the radius is found by EDCORP without any user interaction. Thus, the manual operations of subimage framing, ulna removal, and contour correction required by the current BOFI method used in our Bone Research Unit are seldom required. Even when EDCORP fails to produce an accurate radius contour on its first try, simply removing the ulna from the CT image will, in most cases, allow EDCORP to correctly track the bone contour on a second attempt.

With alterations in detail, the general methods behind EDCORP and LCSKEL could be useful for other contour extraction problems. The methods of EDCORP could probably be used for tracing any edge with sufficient contrast. The strength of cross-correlation of profiles lies in its relative insensitivity to noise compared with other techniques, and the advantage of contour tracking is its unique specification of a single edge.

There may be cases where a ridge finding technique is more appropriate than a true edge detection method. For example, lower sequential thinning could be applied to tracing roads or rivers in aerial photographs, or veins or arteries in angiographs. In such cases, the objects are

already quite thin, and they usually appear continuous; lower sequential thinning would reduce such objects to a single-pixel width contour. In addition, there are many other Mathematical Morphology transformations that are useful for specific applications. The theory of Mathematical Morphology opens up many new prospects for image analysis.

7. Future Developments

While the two newly developed algorithms for contour detection of bone are in many ways already superior to other methods of contour detection, there are several areas where refinements or additions could lead to even better results. Any changes, however, should maintain or improve the degree of automation already present in the algorithms. The tradeoffs for possible improvement in automation, precision, and reliability are further development time and effort, increased programming complexity, and longer run-times. Results of any changes or additions to the algorithms must be investigated not only to ensure correct behavior on the variety of expected images, but also to establish that improvements if any, are worth the extra run-time for the application in question. Also, there is a question as to whether better than ±1 pixel precision in contour detection can be achieved by any digital method.

7.1 Areas of Refinement for EDCORP

Refinements in profile construction, kernel selection, and contour tracking used by EDCORP may lead to marginally more precise and trustworthy contours.

7.1.1 Improving Profile Construction

Profiles could be improved in their sampling of the CT image. Currently, all radial profiles emanating from a single origin at the center of intensity of the bone are

perpendicular to the cortical edge only if the bone is circular. Profiles oblique to the bone edge display a broadened ramp edge; the resulting correlation peak is also broadened. Consequently, noise in the profile is more likely to influence the determination of the edge point. Also, if the bone is not circular, the radial profiles do not sample the bone contour consistently. Sampling is more sparse for portions of the contour further from the origin, thus more interpolation between the determined edge points is required.

Since radius bones are seldom circular, a method for ensuring that all profiles are perpendicular to the bone edge could lead to a more precise contour. One such method would be to use shapes other than a circle to define the location of the profiles (ellipses, for example). The profiles would then be everywhere perpendicular to the outline of this shape and at equally spaced increments along it. Unfortunately, no standard shape describes the radius especially well nor is the size of the radius in a particular image known *a priori*. What is needed is not a standard shape but a reliable first guess for the contour of the bone. Such a guess could be the bone contour currently determined by EDCORP. An iterative method of contour determination using profiles defined perpendicularly to a previous contour may lead to more precise contour detection in cases where EDCORP currently finds a reasonable contour. However, such a method would probably not greatly reduce the

current failure rate since in cases where EDCORP fails, a reasonable first guess would not be supplied.

Also, as it now stands, profile construction in EDCORP samples the image at 0.5 pixel increments along profiles using nearest neighbor interpolation. Better forms of interpolation such as linear or higher degree polynomial fitting would give a smoother profile at any orientation. However, the same degree of smoothness should also be present in the correlation kernel.

7.1.2 Investigating Kernal Selection

The effects of kernel design for use in EDCORP could be further investigated. As previously stated, it seems that only the general character of the kernel is important for good cross-correlation with the profiles. The extent of the kernel in Figure 19(b) was slightly increased and decreased, and different scaling of the kernel's values were considered. No significant change resulted in the contour determination by EDCORP using these various kernels, but only a small set of test images was investigated. The small differences noted may become significant when further alteration to the kernel or other images are studied.

Since the profiles through the cortical bone shell are quite different at various locations along the bone contour, it may be desirable to devise a method that alters the kernel based on information about these profiles.

7.1.3 Refinements to Contour Tracking

The current methods of tracking and backtracking used in EDCORP are quite crude. Starting point selection, search window size selection, and search technique could all be improved.

Improper selection of the first edge point is a problem in EDCORP that seldom occurs. When it does occur, however, the resulting contour is never correct. A possible solution is to choose the outermost 'local' maximum of the cross-correlation values above a certain threshold rather than choosing the overall maximum value. The threshold and the extent of the 'local' window would probably have to be determined by studying a very large number of CT images.

The current edge tracking method in EDCORP was optimized for CT images of right forearms. A different starting profile and tracking direction could be automatically invoked when the CT image is of a left wrist.

In edge tracking, the search window for the next edge point is currently ±5 pixels with respect to the radial location of the previous edge point (±15 pixels for backtracking). This window is too small to properly track around sharp corners, though the erroneous contour often wanders back to the bone edge within a short distance. But it is also too large to avoid mis-tracking when the ulna is very close to the radius. The current backtracking method is not a very robust solution. Perhaps the search window size could be dynamically varied, made smaller at the bottom of

the image when near the ulna and larger at the top right when near the sharp cortical 'cape' that is often present in these CT images. However, a rather sharp corner is also common at the bottom right portion of the radial cortex.

An algorithm similar to EDCORP was independently developed by Pope *et al.* for detection of the left ventrical in angiocardiographs [24]. A major difference in his algorithm is the use of a dynamic search technique for the best edge contour. Consider the bone contour as a path through an edge likelihood matrix whose rows are the cross-correlation functions corresponding to each radial profile. Contour tracking in EDCORP is equivalent to connecting the largest value within the search window of the next row (±5 columns within the column of the current edge point) with the edge point in the current row. Pope considers, effectively, all paths through this edge likelihood matrix. Thus, his search does not depend on starting point selection. The 'score' of each contour is the sum of the cross-correlation values encountered along each path. The path with the highest score is selected as the best bone contour. Dynamic programing is the method which allows for the different contours to be efficiently tracked and scored. This method allows for short lapses in local optimization in order to achieve the best global response. Such a tracking procedure is superior to the simple tracking used in EDCORP. Pope does not, however, discuss what happens when the selected contour is not self-connecting.

7.2 Investigations for LCSKEL

The basic methods of LCSKEL are sound and allow little room for tuning. Probably the most fruitful areas of further study for LCSKEL are threshold selection for the lower clipped skeleton image, and combination with other contour detection methods.

7.2.1 Threshold Selection

A major reason for the occasional failure of LCSKEL in determining an acceptable contour is the automatic threshold selection. The current method of determining the most frequently occurring histogram value between 4000 and 6000 and then subtracting 750 (if this value is over 4750) is quite crude, though it does usually result in a threshold which leaves the main cortical ridge intact while removing the background.

Perhaps a better method may be to choose a somewhat high threshold, and then successively reduce the threshold value until the remaining object is suitably 'closed' with respect to a very large disk-like structuring element. This procedure, however, would require many thresholdings, contour fillings, and closings, employing a great amount of computer processing time.

In any event, it would be useful to include a manual threshold selection option in the modified Post-BOFI viewing procedure. If a particular contour was unacceptable, the operator could choose a lower or higher threshold for the

lower clipped skeleton, which would then be filled and smoothed. This may result in an acceptable contour without resorting to manually 'drawing in' corrections to the contour.

7.2.2 Combining LCSKEL With Other Methods

In Appendix I, it is stated that differencing operators, such as the gradient or Laplacian, tend to enhance all edges within an image. Some sort of pixel classification is then required to extract the relevant contour. Perhaps LCSKEL could be used to detect ridges within such edge enhanced images. This would be especially useful when the original image has no ridges as such or when the edge detection technique produces enhanced edges which are quite broad but not broken.

To investigate these possibilities, a few CT images of the wrist were either 0x0, 3x3, or 9x9 median or average filtered and then edge detected using the 3x3 Sobel compass gradient mask to produce an edge image as discussed in Appendix I. The lower clipped skeletons of these images were determined. Unfortunately, the ridges found by LCSKEL did not correspond well to the bone edge. Most likely, this was due to the noisy nature of the CT images (even with filtering) and the sensitivity of the Sobel method to this noise (although it is one of the better edge operators [37]). However, many other contour detection techniques exist which could be tested in conjunction with LCSKEL.

References

1. Rosenfeld A, Kak AC. **Digital Picture Processing**, 2nd Ed. Academic Press, New York, 1982;2:1.
2. ibid.;2:75-78.
3. ibid.;2:138-145.
4. ibid.;2:145-152.
5. Ibid.;2:89-91.
6. ibid.;2:36-41.
7. ibid.;1:254-261.
8. ibid.;2:232-240.
9. ibid.;2:85-89.
10. ibid.;2:100-101.
11. ibid.;2:241-244.
12. ibid.;2:130-138.
13. Johns HE, Cunningham JR. **The Physics of Radiology**, 4th Ed. Charles C Thomas Publisher, Springfield, Illinois, 1983:411-450.
14. Seitz P, Ruegsegger P. Fast Contour Detection Algorithm for High Precision Quantitative CT. **IEEE Trans Med Phys** 1983;MI-2(3):136-141.
15. Hangartner TN, Overton TR. Quantitative Measurement of Bone Density Using Gamma-Ray Computed Tomography. **J Comput Assist Tomogr** 1982;6(6):1156-1162.
16. Guyton, AC. **Medical Physiology**. WB Saunders Co, Toronto, 1981;Ch 79:973-988.
17. Schulz E, Ferguson J, et al. Bone Density in the Spine by CT and in the Radius by Photon Absorption in

Osteoporosis. **J Comput Assist Tomogr** 1983;7(3):555-556.

18. Rueggsegger P. Development of Quantitative CT Methods for Bone and Soft Tissue Evaluations. **J Comput Assist Tomogr** 1983;7(3):548-549.

19. Brooks RA, Di Chiro G. Principles of Computer Assisted Tomography in Radiographic and Radioisotopic Imaging. **Phys Med Biol** 1976;21(5):689-732.

20. Wells DM. Design, Implementation, and Calibration of a Specialized X-ray CT Scanner for Bone Evaluation. Thesis, University of Alberta, 1986.

21. Davis LS. A Survey of Edge Detection Techniques. **Computer Graphics Image Processing** 1975;4:248-270.

22. Roberts LG. Machine Perception of Three Dimensional Solids. (In) **Optical and Electro-Optical Information Processing**, (J Tippett, D Berkowitz, L Clapp, C Koester, A Vanderburgh, eds), MIT Press 1965:159-197.

23. Keller JM, Edwards MF, Rundle R. Automatic Outlining of Regions on CT Scans. **J Comput Assist Tomog** 1981;5(2):240-245.

24. Pope DL, Parker DL, et al. Left Ventricular Border Recognition Using a Dynamic Search Algorithm. **Radiology** 1985;155(2):513-518.

25. Serra J. **Image Analysis and Mathematical Morphology**. Academic Press, New York, 1982.

26. ibid.:374.

27. Huang TS, Yang GJ, Tang GY. A Fast Two-Dimensional Median Filtering Algorithm. **IEEE Trans Acoust Speech**

Signal Processing 1979;ASSP-27(1):13-18.

28. Dudgeon DE, Mersereau RM. **Multidimensional Digital Signal Processing.** Prentice-Hall, Englewood Cliffs, New Jersey, 1984:15-20.
29. Lathi BP. **Signals, Systems and Communication.** John Wiley and Sons, New York, 1965::515-535.
30. Telford WM, Geldart LP, et al. **Applied Geophysics.** Cambridge University Press, New York, 1976:380-384.
31. Jenkins GM, Watts DG. **Spectral Analysis and its applications.** Holden-Day, London, 1969:322-328.
32. Peli T, Malah D. A Study of Edge Detection Algorithms. **Computer Graphics and Image Processing** 1982;20:1-21.
33. Pratt WK. **Digital Image Processing.** John Wiley and Sons, New York, 1977.
34. Rosenfeld A, Thurston M. Edge and Curve Detection for Visual Scene Analysis. **IEEE Trans Computers** 1971;C-20:562-569.
35. Squires GL. **Practical Physics**, 2nd Ed. McGraw-Hill, New York, 1976:202-204.
36. ibid.:35-37.
37. Shaw GB. Local and Regional Edge Detectors: Some Comparisons. **Computer Graphics and Image Processing** 1979;9:135-149.
38. Duda R, Hart P. **Pattern Classification and Scene Analysis.** John Wiley and Sons, New York, 1973.
39. Prewitt JMS. Object Enhancement and Extraction. (In) **Picture Processing and Psychopictorics**, (BS Lipkin,

A Rosenfeld, eds), Academic Press, New York, 1970:75-149.

40. Kirsch RA. Computer Determination of the Constituent Structure of Biological Images. **Computers and Biomedical Research** 1971;4:315-328.

41. Robison GS. Edge Detection by Compass Gradient Masks. **Computer Graphics and Image Processing** 1977;6:492-501.

42. Rosenfeld A, Thurston, M, Lee YH. Edge and Curve Detection, further experiments. **IEEE Trans Computers** 1972;C-21:677-715.

43. Hayes K, Rosenfeld A. Efficient Edge Detectors and Applications. University of Maryland Tech Rep 207, 1972.

44. Hale HG. Detection of Elementary Features in a Picture by Non-linear Local Numerical Processing. **Proc Third Int Joint Conf on Pattern Recognition** 1976:764-768.

45. Smith MW, Davis WA. A New Algorithm for Edge Detection. **Proc Second Int Joint Conf on Pattern Recognition** 1974:150-151.

46. Hueckel MH. A Local Operator Which Locates Edges in Digital Pictures. **J Assoc Comput Mach** 1971;18:113-125.

47. Hueckel MH. A Local Visual Operator Which Recognizes Edges and Lines. **J Assoc Comput Mach** 1973;20:634-647.

48. Martelli A. Edge Detection Using Heuristic Search Methods. **Computer Graphics and Image Processing** 1972;1:169-182.

49. Montanari U. On the Optimal Detection of Curves in

Noisy Pictures. **Comm Assoc Comput Mach** 1971;14:335-345.

50. Kelly M. Edge Detection by Computer Using Planning. (In) **Machine Intelligence-VI**, (B Meltzer, D Michie, eds), Edinburgh University Press, Edinburgh, 1971:397-409.

Appendix I

Edge detection techniques are often classified as linear or non-linear, parallel or non-parallel, and local or regional.

An operator Ψ is called linear if the same output is obtained whether the operation is applied to a linear combination of component images X_1 ,X_2 that combine to create the image, or Ψ is applied separately to each of the components before combining the results:

$$\Psi\{aX_1 + bX_2\} = a\Psi\{X_1\} + b\Psi\{X_2\},$$

(a,b: gray-level scaling constants).

A parallel technique for edge detection is such that the operator may be applied everywhere in the image simultaneously, at least in principle. The decision of whether or not a set of points belongs to an edge is not influenced by decisions previously made for other sets of points.

Local edge detection methods decide whether or not a point in the image belongs to an edge; they rate the 'edginess' of a point by examining the point and small neighborhoods (generally 3x3 to 9x9 pixels) about the point. In effect, such edge detection operators are combinations of point or neighborhood transformations. This is in contrast to regional edge detection techniques which generally consider a larger region or subimage of the original image and try to find the 'best fitting' edge contour through this region.

Local Edge Detection Techniques

A classical linear, parallel, local operator in image processing is the spatial frequency filter [21]. Since edges are associated with abrupt changes in intensity, the filter should attenuate low spatial frequencies (blurring) and enhance higher ones. However, noise is a higher frequency phenomenon; so the filter should be designed to attenuate such frequencies also. Optimal filter design is an important problem in edge detection [36]. The most common and simplest edge detection techniques are based on differencing: calculating approximate directional derivatives at each point. These are usually parallel and local but may be linear or non-linear. Let X(i,j) be the gray-level at pixel (i,j) and consider the horizontal direction. If the absolute value |Dh| of

$$Dh(i,j) = X(i+1,j) - X(i,j)$$

is high, then there is a good probability that there is a vertical edge between the two pixels considered. |Dh| is a digital approximation to the magnitude of the horizontal directional derivative [9].

Consider also the vertical derivative:

$$Dv(i,j) = X(i,j+1) - X(i,j)$$

a 90° rotation of Dh. The gradient at a given point may be digitally approximated in magnitude GRAD and direction Φ as:

$$GRAD(i,j) = \{(Dh)^2 + (Dv)^2\}^{1/2}$$

$$\Phi(i,j) = \arctan(Dv/Dh).$$

Note that the gradient is always orthogonal to the edge

direction; a convention such as 'brighter pixels to the left' is usually adopted to uniquely specify the edge direction. The magnitude of the gradient is taken as the strength of the edge.

For continuous functions, given derivatives D1,D2 in any two non-collinear directions, the derivative in any desired direction may be expressed as a linear combination of them. Thus, other orthogonal directions are often used. For example, Roberts [22] approximates the magnitude of the gradient by using diagonal differences:

$$GRAD(i,j) = \{(X(i,j)-X(i+1,j+1))^2+(X(i+1,j)-X(i,j+1))^2\}^{1/2}.$$

It is common to use some variation in combining the two responses D1,D2 used to approximate the gradient to avoid the computational cost of the square root. For example, the sum of responses may be used [21,37]:

$$GRAD(i,j) = |D1| + |D2|.$$

This is an over-estimate to the norm in the previous equation. An under-estimate is given by [32]:

$$GRAD(i,j) = \max\{|D1|,|D2|\}.$$

It is also common to obtain a directional derivative at each point by convolving the image with a mask or template. For example, the mask for a vertical edge (horizontal gradient) is:

$$\mathbf{-1}\quad +1$$

where the bold indicates the masks origin. Symmetrical masks are often used:

$$-1\quad \mathbf{0}\quad +1 \qquad \text{vertical edge } (270°)$$

$$\begin{matrix} +1 \\ 0 \\ -1 \end{matrix}$$ horizontal edge (0°).

The output of these symmetrical masks represents the derivative at the central pixel, instead of between two pixels, but ignores the central pixel in the calculation.

In calculating the gradient using the above masks, only four points are weighted making this operation extremely sensitive to noise. Such masks are often extended to include one- or two- dimensional neighborhoods on either or all side(s) of the point for which the gradient is being calculated as shown in Figure 29. Other examples of extended masks are given by Prewitt [39]:

$$\begin{matrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{matrix}$$

by Sobel [38]:

$$\begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}$$

and by Kirsch [40]:

$$\begin{matrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{matrix}$$

All may be extended to larger (usually up to 9x9) window sizes (see also the Hummel operator [37]). All three 3x3 masks are a combination of spatial averaging and differentiation. They may be considered examples of 2-D spatial frequency filters previously discussed; so filter optimization is again involved.

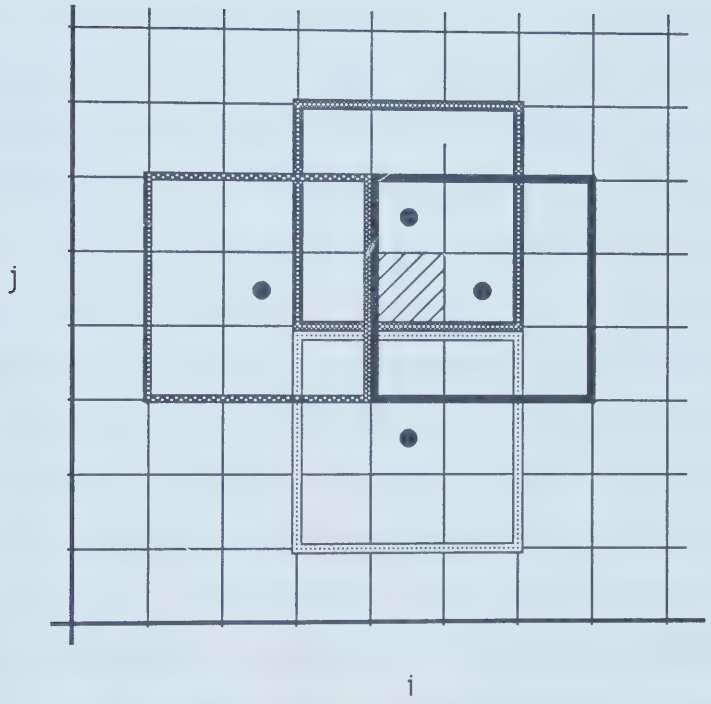
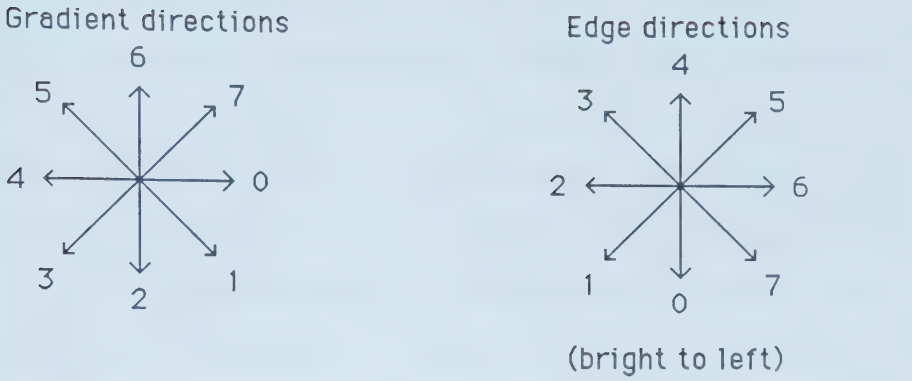


Figure 29. Possible one- and two-dimensional neighborhoods about a point used in calculating the gradient.

Since images are not continuous but discrete, considering two or even four directional derivatives to estimate the gradient is not sufficient. Thus, eight directions are often considered and the maximum response among the masks comprising this directional set is used to determine the gradient. For example, Figure 30 shows the eight compass gradient masks for the Sobel design. The eight gradient directions or the corresponding eight edge directions (brighter pixels to left) are encoded from 0 to 7 as shown. The maximum output value from the eight masks applied to a pixel in the image is written to the corresponding pixel in the edge (or gradient) magnitude map, and the direction code for the mask giving this maximum value is written to the same pixel location in the edge (or gradient) direction map.

Depending on how the responses from the masks of the various directions are combined for each point, the above local differencing operators may be linear (if the actual gradient or edge vectors are encoded as norm and direction) or non-linear (if, for example, only the gradient magnitude or maximal response is calculated). Sometimes logical conditions are included into the definition of the mask response (for example, requiring that each '1' pixel in the Prewitt mask be greater than the corresponding '-1' pixel [10]) or into the response of neighboring points (for example, requiring that the gradient direction associated with a pixel be compatible with the directions of the



Compass direction				Gradient	Edge
E	-1	0	1		
	-2	0	2		
	-1	0	1	0	
SE	-2	-1	0		
	-1	0	1		
	0	1	2	1	
S	-1	-2	-1		
	0	0	0		
	1	2	1	2	
SW	0	-1	-2		
	1	0	-1		
	2	1	0	3	

Compass direction				Gradient	Edge
W	1	0	-1		
	2	0	-2		
	1	0	-1	4	
NW	2	1	0		
	1	0	-1		
	0	-1	-2	5	
N	1	2	1		
	0	0	0		
	-1	-2	-1	6	
NE	0	1	2		
	-1	0	1		
	-2	-1	0	7	

Figure 30. The eight compass directions for the Sobel compass gradient masks.

preceding and succeeding pixel [41]). Such non-linear modifications can be incorporated into many edge detection methods.

Another linear, local edge enhancement technique is to compute the Laplacian or 2-D second derivative at each point [11]. If -1 +1 approximates the first derviative just to the right of a pixel and -1 +1 just to the left, then their difference should approximate the second derivative at the point:

1 -2 1.

The Laplacian operator is the combination of this mask in two dimensions:

$$\begin{matrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{matrix}$$

This mask can be extended to larger sizes. Again, it is simply a 2-D spatial filter. Theoretically, the zero crossing between a negative and positive value of the Laplacian image, produced by convolving this mask with the image, indicates an edge. Sometimes, edges are enhanced by subtracting the Laplacian image (LAP) from the original image (unsharp masking) [11]. Pixels just below an edge will then be reduced in value (since here, LAP>0), and pixels just above the edge will be increased (LAP<0). Points away from the edge will be left unaltered (LAP≃0). The Laplacian, though, is extremely sensitive to noise and is not orientation invariant [5].

Non-Linear, Local Edge Detection Techniques

Non-linear, local edge detection techniques include those by Rosenfeld [21,32,34,42,43], Hale [32,44], Smith and Davis [32,45], and Herskovitz and Binford [21]. These are generally extensions to or variations of differencing techniques. The best of these [32] are due to Rosenfeld; so only they will be discussed.

Noise greatly influences the differencing methods of edge detection since a bright pixel in the neighborhood of a point may erroneously lead to the detection of an edge at that point, especially if the neighborhood is small. The differencing techniques also detect edges rather sloppily since they give a high value not only on the edge but, depending on the size of the neighborhoods considered, also at points farther from the actual edge. Rosenfeld's methods of edge detection try to overcome these problems by using many neighborhoods of various sizes [42]. At each pixel (i,j) the differences between average gray-levels of pairs of 2-D neighborhoods (like those in Figure 30) of all sizes 1x1 to mxm are calculated. Let E_{ij}^{k} be one such difference using kxk pixel neighborhoods.

One of Rosenfeld's algorithms for edge detection computes the products M(i,j) at each pixel:

$$M(i,j) = \prod_{k=1}^{m} E_{ij}^{k}$$

This product will be large only if all the E_{ij}^{k} are large [34]. The effects of noise are reduced by using large neighborhoods while small neighborhoods quickly reduce the

value M(i,j) away from the actual edge.

In another of Rosenfeld's methods, a 'largest size' neighborhood for each point is calculated such that E_{ij}^{k} is not significantly smaller than E_{ij}^{k-1}. Strong edges are then sharply determined by computing:

$$S(i,j) = E_{ij}^{m}$$

if $E_{ij}^{m} \geq E_{i'j'}^{m'}$, for all (i',j') within m/2 of (i,j),

$$= 0 \text{ otherwise,}$$

where m and m' are the 'largest sizes' at (i,j) and (i',j'), respectively. By only marking the strongest edge within a small distance of each point, minor edges due to noise near major edges are ignored, while isolated low contrast edges from actual objects will be detected [43].

Regional Edge Detection Methods

In contrast to local edge detection operators which are applied to each pixel and give a measure of the edge likelihood of that pixel, regional edge detection techniques usually consider overlapping regions of the image and try to find the 'best' edge contour through this region. In this sense, the first of the previously discussed Rosenfeld algorithms may be considered a regional method.

A famous regional edge detection algorithm was introduced by Hueckel [37,46,47]. This method uses overlapping, approximately disk shaped regions or windows, a set of nine orthogonal basis functions, and a generalized ideal edge function. Both the portion of the image in each

disk-shaped window and the ideal edge are expanded in terms of these basis functions. The error (sum of squared differences between corresponding expansion coefficients) between the image and the ideal edge is minimized with respect to the parameters describing the ideal edge function. This gives a solution for the specific ideal edge best matching the portion of the image within the window. If the error is sufficiently small, then the ideal edge is taken as defining the edge contour through that region. Overlapping the disk shaped regions ensures continuity. Also, for each edge contour, a 'goodness of fit' measure is provided so that one may weight the results appropriately.

Hueckel's method of detecting edges is based on a continuous (analogue) class of edge models so that usual analytic optimization procedures can be employed. Shaw, however, suggests using a regional edge fitting algorithm based on intrinsically discrete edge models and employing rectangular, overlapping windows [37]. His edge model is a template where each row of the template contains a left/right division into a dark and bright side. To represent an ideal edge, such division points are consistent in their handedness and vary in a smooth way from row to row. Shaw chooses monotonicity from top to bottom (or left to right) of these edge division points as his criterion for smoothness. His algorithm attempts to fit a template of this type to the portion of the image within each window. The division points of the template with the required smoothness

which best fits the image then define the edge contour through that window.

Non-Parallel Edge Detection Methods

All of the above edge detection techniques are considered parallel methods of edge detection. The regional edge operators do consider previously determined edge points within their window when constructing the best edge contour through that window, but each window, and thus each edge contour, is considered separately. The solution for each window is only locally optimal. For non-parallel or sequential detectors, the decision of choosing the next set of points which belong to an edge is contingent upon the results of the edge operator at previously considered points. In many ways, the most popular non-parallel edge detecting techniques can be considered similar to those regional methods previously discussed but with a single window extended to include the entire image.

Examples of non-parallel edge finding methods include heuristic search procedures [21,48], dynamic programming [24,49], edge tracking [12], and guided edge detection [50].

Heuristic search procedures seek a minimal cost path through a state space from an inital state to some goal or termination state using a cost function. For edge detection, these states or nodes may be edge elements between high contrast pixels, and the cost associated with a state may be an inverse measure of this contrast (being low cost for high

contrast states). The cost function for the 'nth' node in the path may be the sum of costs to get to state 'n' from the initial state plus an estimate of the cost to continue on to the termination state. The search for an edge contour is guided by choosing those states with minimum cost function.

Dynamic programming, in these terms, calculates the cost function for all paths through the state space, within certain restrictions, rather than using it to guide the search. The best path is then chosen as the one with the overall minimum cost. In exchange for more computation, dynamic programming allows for short lapses in optimization. The chosen edge contour may not be described by the best edge at every location but is instead the best contour overall.

Edge tracking employs tracking criteria based on gray-level information, gradient magnitude and direction. The image is first scanned to find starting points that meet the detection criteria (strong edge points). Neighborhoods around these current points are analyzed to select succeeding points belonging to the edge being tracked. In some algorithms, the tracking criteria at any location may depend on the gray-level, contrast, or other proerties of the current edge point or previous points. The tracking criteria may also be updated based on properties of the candidate pixels, including their distance and direction from the current edge point. This allows the algorithm to

tolerate small gaps but bias against points far from the current point and to accept points on a smooth curve but not those that deviate too much from the tracking direction. Ususally, the candidate pixel which meets the criteria and is closest to the current point is accepted as part of the edge and becomes the new current point. Tracking continues, for closed objects, until the edge returns to the starting point or runs off the image border. Several extensions to edge tracking methods have been developed such as the ability to 'backup' if a branch seems to be leading to poorer acceptances.

Guided edge detection schemes make use of a plan or model as a guide. The model may be built into the program, input by a user, or determined by the program itself. The program is guided by the model as to where edges should be, what characteristics they possess, and how they are connected. This model is often updated and then used to help guide the algorithm to iteratively determine other or more accurate edges. For example, guided edge detection takes place in edge tracking when the tracking criteria for succeeding pixels are based upon properties of the current or candidate edge points.

Edge Selectiveness

The previous examples of edge detection methods have been presented in a generally increasing order of contour selectiveness or tunability. The local edge detection

techniques (spatial frequency filtering, differencing techniques, Rosenfeld's first method), to some degree, enhance all edges within the image, whether they are real or due to noise. If a certain edge is desired, it must somehow be extracted, for example, by suitably thresholding the edge enhanced image or using some other form of pixel classification.

Rosenfeld's second method zeros all but the strongest edge in small overlapping regions. Similarly, the regional edge operators (Hueckel, Shaw) find only one edge in each region. The basis functions are defined to have maximum overlap with edges of a particular character.

Most of the non-parallel edge detection methods (heuristic search, dynamic programing, edge tracking, but not necessarily guided edge detection) are even more selective, usually determining only one or a few (best) contours at a time in the entire image. In the edge tracking procedure, for example, the definition of the tracking criteria determines which closed object contour in the image will be followed.

If detection of a specific edge or contour in the image is desired, then the edge detection technique should be tuned to utilize the particular characteristics of that edge. However, as variations among images of a particular class increase, the contour detection algorithm must be less selective or else be 'self-tuning' to accept the variations in contour parameters.

Appendix II

Algebraic properties of operations in Mathematical Morphology can be proven using set theory. As an example of the algebraic manipulations available in Mathematical Morphology, the proofs of duality with respect to complementation, iterativity (or distributivity), and frame erosion are presented [25].

Duality With Respect to Complementation

Duality with respect to complementation of erosion and dilation means that, for example, complementing the erosion of X by B results in the same image as dilating X^C by B, and conversely. Recall the mathematical definition of the erosion:

$$(X \ominus \check{B}) = \bigcap_{mn \epsilon \check{B}} X_{mn}$$

So:

$$\begin{aligned}(X \ominus \check{B})^C &= \left(\bigcap_{mn \epsilon \check{B}} X_{mn}\right)^C \\ &= \bigcup_{mn \epsilon \check{B}} X^C_{mn} \\ &= (X^C \oplus \check{B})\end{aligned}$$

by the mathematical definition of the dilation.

Iterativity

The property of iterativity allows complex structuring elements to be represented by a dilation (sum) of simpler component structuring elements. For example, a square structuring element with sides of length 's' can be considered as the dilation of a vertical line of 'on' pixels

(B) with a horizontal line of pixels (B'), both of length 's'. The dilation of a set X by such a structuring element (B ⊕ B') is equivalent to iteratively dilating X by the simpler component structuring elements.

In general:

$$\begin{aligned}(X \oplus \check{B}) \oplus \check{B}') &= \bigcup_{mn\epsilon\check{B}'} (X \oplus \check{B})_{mn} \\ &= \bigcup_{mn\epsilon\check{B}'} \bigcup_{pq\epsilon\check{B}} X_{p+m,q+n} \\ &= \bigcup_{kl\epsilon(\check{B}\oplus\check{B}')} X_{kl} \\ &= X \oplus (\check{B} \oplus \check{B}').\end{aligned}$$

The proof for iterativity of erosion is similar.

Frame Erosion and Local Knowledge

Local knowledge due to the finite nature of some images means that the set X is known only inside some frame or window Z. When an erosion or dilation is performed on X, one does not know how portions of X outside the frame will be affected nor how these portions will affect results inside the frame Z. The solution is to limit the frame of the result to what is positively known locally.

For erosions, the procedure is obvious. Since one knows only $X \cap Z$, and since

$$\begin{aligned}(X \cap Z) \ominus \check{B} &= \bigcap_{mn\epsilon\check{B}} (X \cap Z)_{mn} \\ &= \bigcap_{mn\epsilon\check{B}} (X_{mn} \cap Z_{mn}) \\ &= \left(\bigcap_{mn\epsilon\check{B}} X_{mn}\right) \cap \left(\bigcap_{mn\epsilon\check{B}} Z_{mn}\right) \\ &= (X \ominus \check{B}) \cap (Z \ominus \check{B}),\end{aligned}$$

it is seen that during erosion of X, the frame Z must also be eroded by the same structuring element.

The corresponding procedure for dilation is not as straight forward. Consider the (known) set $(X \cup Z^c) = (X \cap Z) \cup Z^c$. No matter how X is altered outside of Z, this set is left unchanged. Note also that:

$$\begin{aligned}(X \cup Z^c) \oplus \check{B} &= \bigcup_{mn \in \check{B}} (X \cup Z^c)_{mn} \\ &= \left(\bigcup_{mn \in \check{B}} X_{mn} \right) \cup \left(\bigcup_{mn \in \check{B}} Z^c_{mn} \right) \\ &= (X \oplus \check{B}) \cup (Z^c \oplus \check{B}) \\ &= (X \oplus \check{B}) \cup (Z \ominus \check{B})^c\end{aligned}$$

The points $(Z \ominus \check{B})^c$ are of no interest and should be excluded; so we note that the largest frame whose intersection with $(Z \ominus \check{B})^c$ is empty is $(Z \ominus \check{B})$. Therefore,

$$[((X \cap Z) \cup Z^c) \oplus \check{B}] \cap (Z \ominus \check{B}) = (X \oplus \check{B}) \cap (Z \ominus \check{B})$$

is the largest dilation which is positively known for any general X. The frame $(Z \ominus \check{B})$ is the largest inside which the dilation of X by B is positively known independent of X.

Thus, to dilate or erode set X by B, local knowledge requires that the frame inside which X is known must be eroded by the same structuring element B.

B45633